NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

THE OPTIMIZATION OF A DUAL FOIL FLAPPING DEVICE

by
Craig J. Paganucci

September 2003

Thesis Advisor: Kevin D. Jones
Second Reader: Max F. Platzer

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: The Optimization of a Dual Foil Flapping Device			**5. FUNDING NUMBERS**
6. AUTHOR(S)			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Aquatic animals, such as dolphins and tuna, have the ability to swim and maneuver at much greater capacity than any man-made device. If their propulsion methods could be replicated mechanically, the benefits to underwater propulsion would be great. A dual foil pitching-plunging device is used to replicate the basic swimming motion of a dolphin. Numerical simulations are used to predict the behavior of a single foil configuration and its wake. The numerical results are used to predict the behavior of the device and to better direct the experimental study. Experimentally, both a single and dual foil configuration are optimized, with the goal being to determine the optimal conditions for maximizing aft foil thrust production.

14. SUBJECT TERMS Flapping wing propulsion, Underwater propulsion, Navier Stokes, Numerical Prediction			**15. NUMBER OF PAGES** 73
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

THE OPTIMIZATION OF A DUAL FOIL FLAPPING DEVICE

Craig J. Paganucci
Ensign, United States Naval Reserve
B.S., Georgia Institute of Technology, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author: Craig J. Paganucci

Approved by: Kevin D. Jones
Thesis Advisor

Max F. Platzer
Second Reader

Max F. Platzer
Chairman, Department of Aeronautics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Aquatic animals, such as dolphins and tuna, have the ability to swim and maneuver at much greater capacity than any man-made device. If their propulsion methods could be replicated mechanically, the benefits to underwater propulsion would be great. A dual foil pitching-plunging device is used to replicate the basic swimming motion of a dolphin. Numerical simulations are used to predict the behavior of a single foil configuration and its wake. The numerical results are used to predict the behavior of the device and to better direct the experimental study. Experimentally, both a single and dual foil configuration are optimized, with the goal being to determine the optimal conditions for maximizing aft foil thrust production.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to thank Dr. Kevin Jones for his help. Without Dr. Jones' assistance and insight, this thesis would have not been possible. Also, I would have never known about the possibility of this research if Chris Bradshaw hadn't informed me of it.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. OVERVIEW

For centuries, mankind has been in awe of the speed and grace with which dolphins swim, watching them from the decks of ships as the dolphins surf the bow waves of ships. In these displays, dolphins keep pace with ships traveling at 20 knots and dart in and out of the bow waves and wakes [Ref 1]. Dolphins are not alone in their swimming ability; yellow fin tuna can manage bursts of speed up to 40 knots [Ref 2]. Speed is not the only advantage enjoyed by these animals, as they are also highly maneuverable, with some fishes being able to reverse direction without any loss of speed.

A mechanical system that could replicate even a fraction of the natural thrust and efficiency of aquatic animals would be far superior to any current method of aquatic propulsion, with applications in both surface and underwater propulsion. Such a device is not beyond the reach of the current state of the art in propulsive technology.

This paper intends to investigate such a possibility through the use of a dual foil plunge/pitch device. Performance of a single foil will be predicted numerically using a Navier-Stokes Solver. These results will then be applied to the device in a series of water tunnel experiments designed to maximize the thrust produced by the device.

B. BACKGROUND

Flapping-wing propulsion was first investigated independently by Knoeller [Ref 3] in 1909 and Betz in 1912[Ref 4]. Their research found that a flapping airfoil in a free stream creates an effective angle of attack, which in turn generates a force vector composed of normal components of lift and thrust, as seen in Figure 1. This effect was experimentally verified by Katzmayr [Ref 5] in 1922. In wind tunnel tests, an oscillating wind stream across a stationary airfoil produced a thrust force. Birnbaum [Ref 6 and 7], in between 1924 and 1925 did extensive work in this field, producing a solution for incompressible flow past a flapping airfoil and measuring the conditions, which lead to flutter.

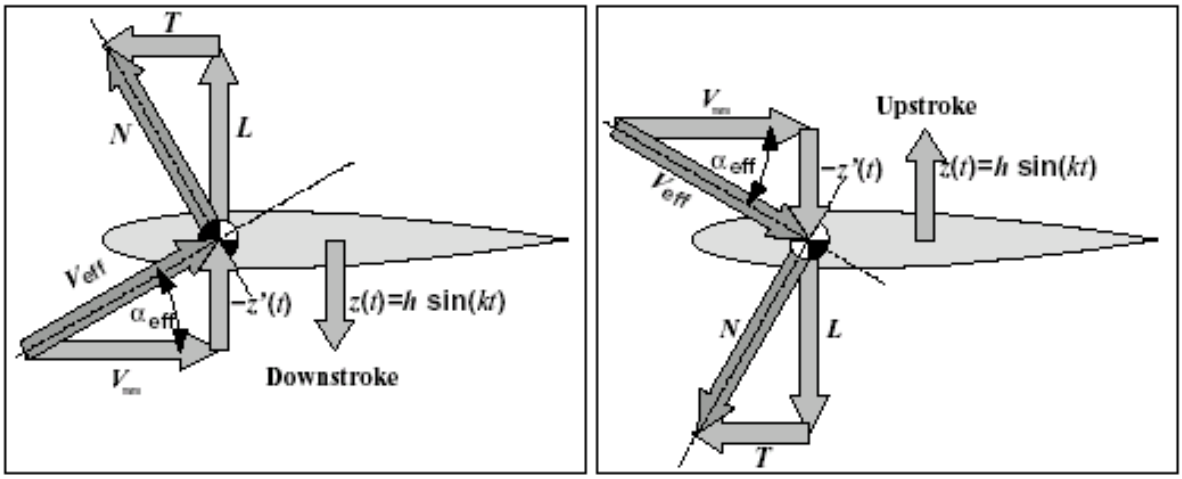


Figure 1. Knoeller – Betz Effect [Ref 8]

As shown in 1935 by Von Karman and Burgers [Ref 9], an airfoil subject to oscillatory movement generates vortex streets. The nature of these vortex streets is dictated by the Strouhal number, defined by Equation 1

$$Sr = \frac{2fch}{U_\infty} \tag{1}$$

where f is the frequency of the airfoil's movement, c is the dimensional chord length, h is the non-dimensional plunge amplitude, and U_∞ is the free stream velocity.

For an airfoil oscillating at a low Strouhal number, the streets are Karman vortex streets, as seen in Figure 2. These vortices are indicative of a drag producing case, where the normal force vector, N, is directed towards the trailing edge of the airfoil, it being made up of a lift, L, and a drag, D, component. Airfoils with a higher Strouhal number generate vortex streets such as those seen in Figure 3. In these cases, the normal force vector is canted forward, being the resultant of a lift and thrust vector.

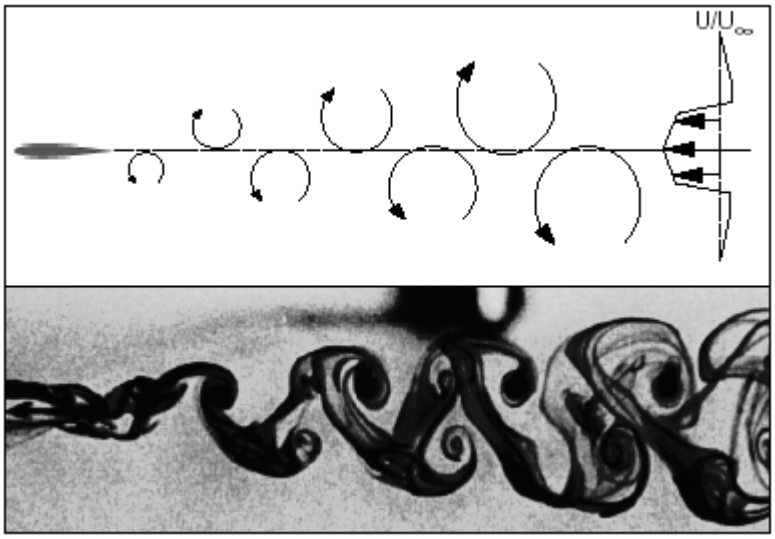


Figure 2. Drag indicative Von Karman Vortices [Ref 10]

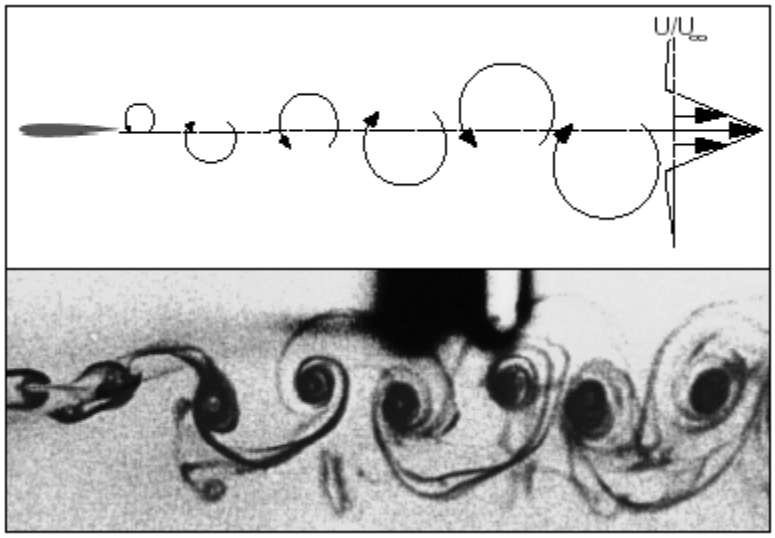


Figure 3. Thrust indicative Von Karman Vortices [Ref 10]

Garrick [Ref 11], in 1936, expanded Theodorsen's theory in his investigation of pitching and plunging airfoils. Schmidt, building on Katzmayr's research, built his *wave propeller* [Ref 12]. The tandem foil *wave propeller,* as shown in Figure 4, functioned by having an oscillatory forward foil and a stationary rear foil. The rear foil captured some of the energy of the vortices produced by the oscillating forward foil, thereby creating additional thrust. Since the rear foil required no energy, the efficiency of the system was increased.

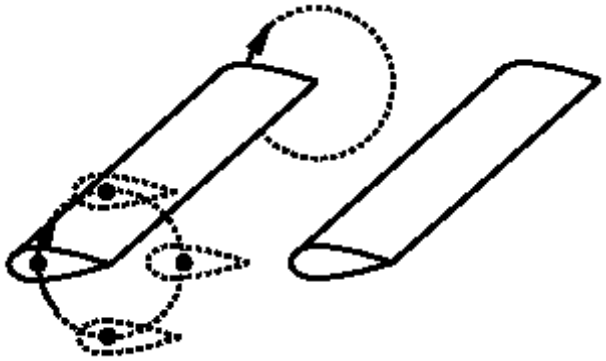


Figure 4. Schmidt wave propeller [Ref 10]

More recently, the use of oscillating foils for high propulsive efficiencies has been researched by a group from the Massachusetts Institute of Technology [Ref 2]. Their work has culminated in the RoboTuna; a robotic fish which uses its tail fin to extract energy from both ambient vortices and vortices shed by the nose and dorsal fin. Through a system of pressure sensors, the ambient vortices are detected, and the tail is maneuvered to utilize the energy found in the vortex.

This paper intends to look at the synthesis of these two ideas by using a dual foil mechanism. This device is intended to provide a simplified version of the movements of aquatic animals, such as dolphins. The forward foil uses an oscillatory motion to generate vortices in its wake. The rear foil will then be positioned to obtain the maximum energy from the vortices in the wake. By using two active foils, a maximum thrust will be produced for work into the device, thereby maximizing efficiency.

C. THEORY

An airfoil that oscillates in both pitch and plunge and has an arbitrary phase angle between the two motions can be described by the following equations:

$$\alpha_g(\tau) = \Delta\alpha_g \sin(k\tau + \phi) \tag{2}$$

and

$$y(\tau) = h \sin(k\tau) \tag{3}$$

where τ is non-dimensional time, h is the non-dimensional plunge amplitude, $\Delta\alpha_g$ is the pitch amplitude, ϕ is the phase angle between pitch and plunge, and k is the reduced frequency, as given by Equation 4,

$$k = \frac{2\pi fc}{U_\infty} \tag{4}$$

where c is the chord length of the airfoil, U_∞ is the free stream velocity, and f is the frequency of oscillation in Hz. Equations 2 and 3 are simplified versions of the airfoil's motion used in effective angle of attack calculations. The foils actually move through a circular arc due to the radial arm that the foils are attached to. Both the NS code and the experiment include an $x(\tau)$ to account for this motion. The thrust coefficient generated by the oscillation of the airfoil is shown in Equation 5

$$C_T = \frac{T}{1/2\ \rho U_\infty^2 S} \tag{5}$$

where, S, is the wing area.

The effective angle of attack is the difference between the induced angle of attack and the geometric angle of attack, as show in Figure 5.

Induced angle of attack is generated by the plunging motion of the airfoil and is given by Equation 6 if the pivot point is at the leading edge; otherwise it is an approximation of the induced angle of attack.

$$\alpha_i(\tau) = \arctan\left(\frac{hk\cos(k\tau)}{U_\infty}\right) \tag{6}$$

Effective angle of attack is given by

$$\alpha_e(\tau) = \alpha_g(\tau) - \alpha_i(\tau) \tag{7}$$

where α_g is the geometric angle of attack given by Equation 2.

Figure 6 illustrates the difference between effective and geometric angle of attack. 6a illustrates a zero geometric angle of attack. Due to the oscillation from the plunge

motion, there is a sinusoidally changing effective angle of attack. 6b shows a case with no induced angle of attack, but with a changing geometric angle of attack.

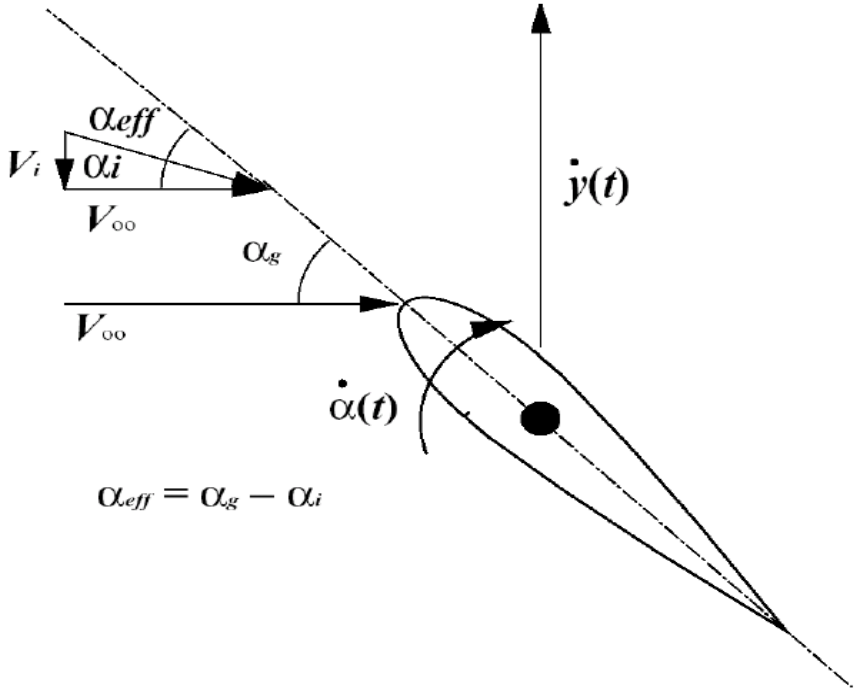


Figure 5. Angles of Attack

6c illustrates the feathered case, where induced and geometric angle of attack cancel each other such that $\alpha_e = 0^o$. 6d shows a case where the geometric angle of attack is less than the induced angle of attack, resulting in thrust production. Finally, 6e illustrates a power extraction case, with the geometric angle of attack exceeding the induced angle of attack.

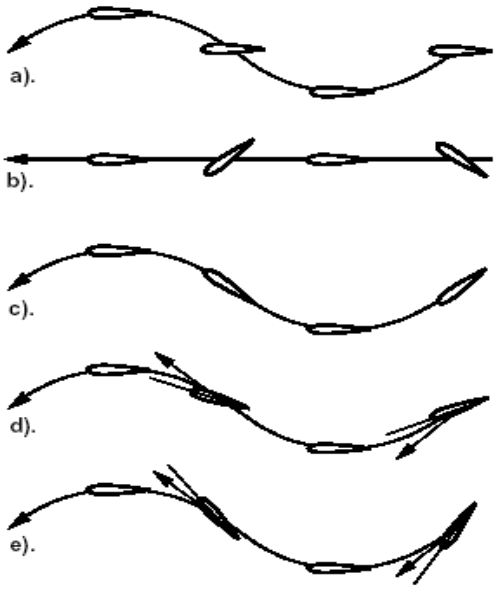


Figure 6. Effective versus geometric angle of attack [Ref 10]

THIS PAGE INTENTIONALLY LEFT BLANK

II. NUMERICAL ANALYSIS

A. UPOT

UPOT, or Unsteady Potential Code, initially developed by Teng, has undergone numerous revisions by both Dr. Platzer and Dr. Jones, as well as others. UPOT is a panel method based on potential flow theory that has been expanded to allow for solving non-linear unsteady flows that are the result of vortex shedding from an oscillating airfoil [Ref. 14]. Panel methods, including UPOT, are limited by the fact that they use the solution to the Laplace equation. The Laplace equation assumes inviscid, irrotational, incompressible flow modeled by summing simple flows. This causes inherent limitations to the applicability of the code. UPOT simulations, however, are very cost effective, providing solutions, albeit limited ones, at a low expense. As such, UPOT was utilized to provide a broad survey of several parameters to indicate general trends. Also, it was used to provide prediction data to complement the Navier-Stokes data.

B. NAVIER STOKES

Navier-Stokes (NS) solvers use fewer assumptions than do panel codes resulting in more accurate predictions. An example of this is shown in Figure 7. Illustrated here is the solution of flow over a steady airfoil at low Reynolds numbers. Clearly visible is the vortex street being shed by the airfoil. This phenomenon is the result of accurate modeling of the viscous effects by the NS solver at low Reynolds numbers. This view of the airfoils is impossible to replicate in UPOT, due to its inviscid flow limitations.

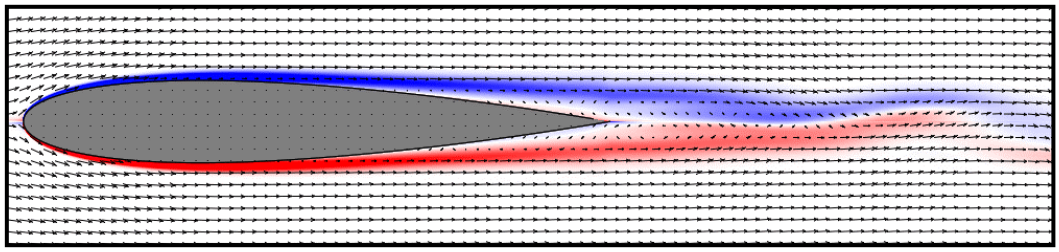


Figure 7. NS steady state solution of an airfoil

The downside to the NS solver is that while the predictions are more accurate, they are much more expensive, each solution taking on the order of 24 hours to compute. The solver works by using a variation of the NS equations; finite differences are used in place of the differential equations. The finite differences are resolved on a grid, the size of which dictates the accuracy of the results. Coarser grids will allow the program to run faster, but real world phenomena may not be resolved due to numerical viscosity.

Several assumptions are made by the NS solver to enable faster times to a solution. The first is the thin-layer approximation, which says that the streamwise viscous terms of the NS equations are omitted. This omission is justified by the fact that the viscous terms normal to the flow are an order of magnitude larger than the streamwise terms. Since the grid is too coarse to resolve the turbulent fluctuations, Reynolds averaging is used for high Reynolds number flows and a turbulence model is used to predict the effect of flow turbulence.

The goal of the NS portion of this study was to find the parameters at which optimum propulsive efficiency is achieved and to estimate interference effects and optimal phasing. Propulsive efficiency is defined as:

$$\eta_{prop} = \frac{\text{Power Out}}{\text{Power In}} = \frac{TU_{\infty}}{P} = \frac{C_T}{C_P} \tag{8}$$

To achieve this objective, a manual optimization procedure was adopted. Propulsive efficiency was chosen over coefficient of thrust as the target variable due to the goal of this paper, i.e. if a fish has a given strength, how does it maximize thrust? Large values of C_T would not necessarily be the most efficient operating point for the airfoil. However, in retrospect C_T would have been the best choice of target variable as it was impossible to experimentally measure the propulsive efficiency of the device.

The optimization process began by running a sweep of reduced frequency values at several different effective angles of attack and plunge amplitudes. The point of highest efficiency was then further resolved by using finer increments for angle of attack. A sweep of airfoil pivot locations was run to find the optimum pivot position. Finally, a sweep of phase angles was performed. Additionally, other cases were investigated, such as a feathered case, a pure plunge case, and also several cases where the airfoil was

moved to a position forward of the device's mast, as illustrated in Figure 8, and compared to similar positions in the standard configuration with the foil aft of the mast.

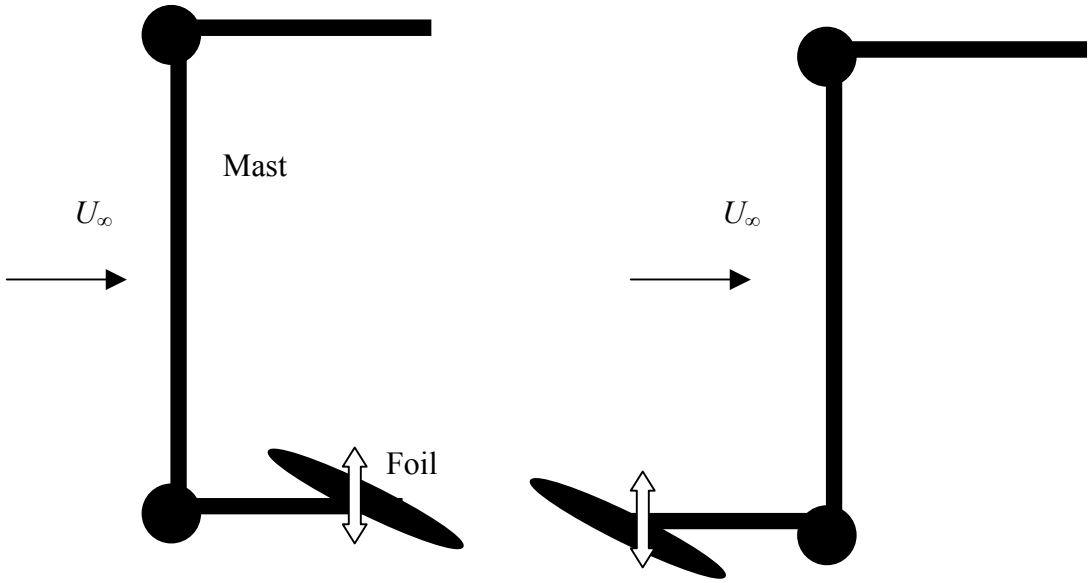


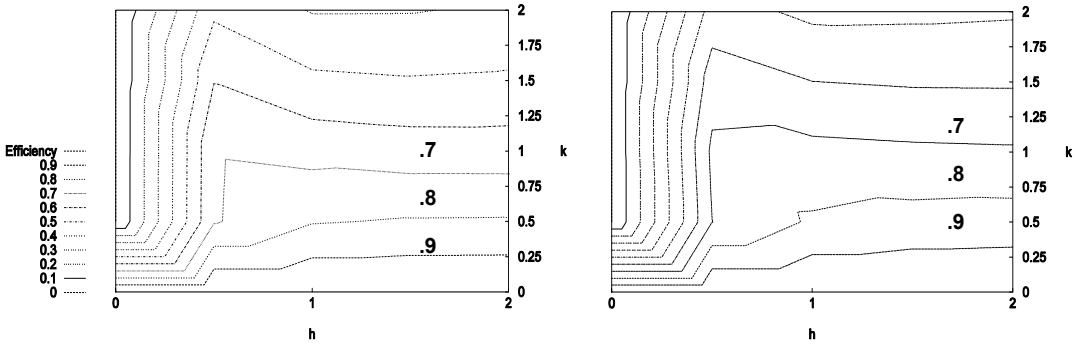
Figure 8. Foil Configuration comparison

THIS PAGE INTENTIONALLY LEFT BLANK

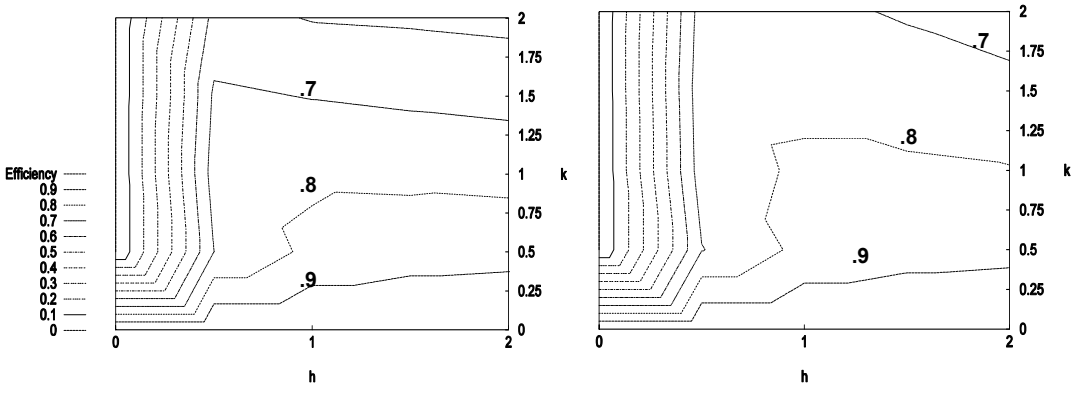
III. NUMERICAL RESULTS

A. UPOT RESULTS

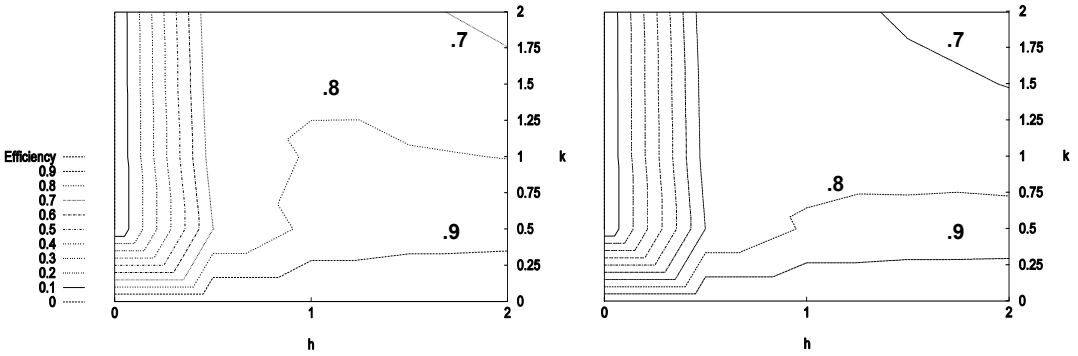
UPOT was used to produce efficiency maps across varying values of *h* and *k*. While not utilized during the NS optimization process, these maps were useful in the experimental procedure. The maps, shown in Figure 9, were used to provide a quick prediction for how efficiency changed due to phase angle, *h,* and *k*. These maps were produced using a NACA 0014 airfoil with an effective angle of attack of 15°. This angle of attack was chosen to give realistic results. Pitching was about the .375 chord point. The figures show that UPOT predicts increasing efficiency with increasing phase angle, with peaks at around 115°.



ϕ=80° ϕ=90°



ϕ=100° ϕ=110°



ϕ=120^o ϕ=130^o

Figure 9. UPOT efficiency maps

B. NAVIER-STOKES OPTIMIZATION RESULTS

By following the procedure described in the previous section, an optimum airfoil setting was found for a single plunging airfoil. First, the airfoil was run through several values of reduced frequency, k, for two different values of plunge amplitude, h, and effective angles of attack, α_{eff}. These initial runs were all carried out with an airfoil pivot point of .250c. The results for each plunge amplitude are shown in Figures 10 and 11.

Several trends become readily apparent from these graphs. First, a larger plunge amplitude results in a higher efficiency. This is not necessarily expected, as a larger h value will usually increase C_T, but thrust and efficiency are often inversely related. Also, a smaller value of k results in a much higher efficiency, with maximum peaks being found about k=.5 for h=1.3. Finally, α_{eff} has a clear effect on the efficiency of the flapping. A smaller α_{eff} results in higher efficiencies. A large α_{eff} causes unfavorable dynamic stall, resulting in reduced efficiency.

The next step in the optimization procedure was to further vary the α_{eff} with the k that produced the peak efficiency. Figure 12 shows the results of varying α_{eff} with a k of .5 and an h of 1.3.

A trend is clearly visible at the lower angles of attack. By reducing the α_{eff}, the airfoil is kept from stalling, thereby resulting in a favorable efficiency. Flow visualization, illustrated later, at this state reveals the lack of airfoil stall.

Finally, the pivot location of the airfoil was varied for the peak case. The results are shown in Figure 13. This figure clearly illustrates the pivot location necessary for peak efficiency in the airfoil

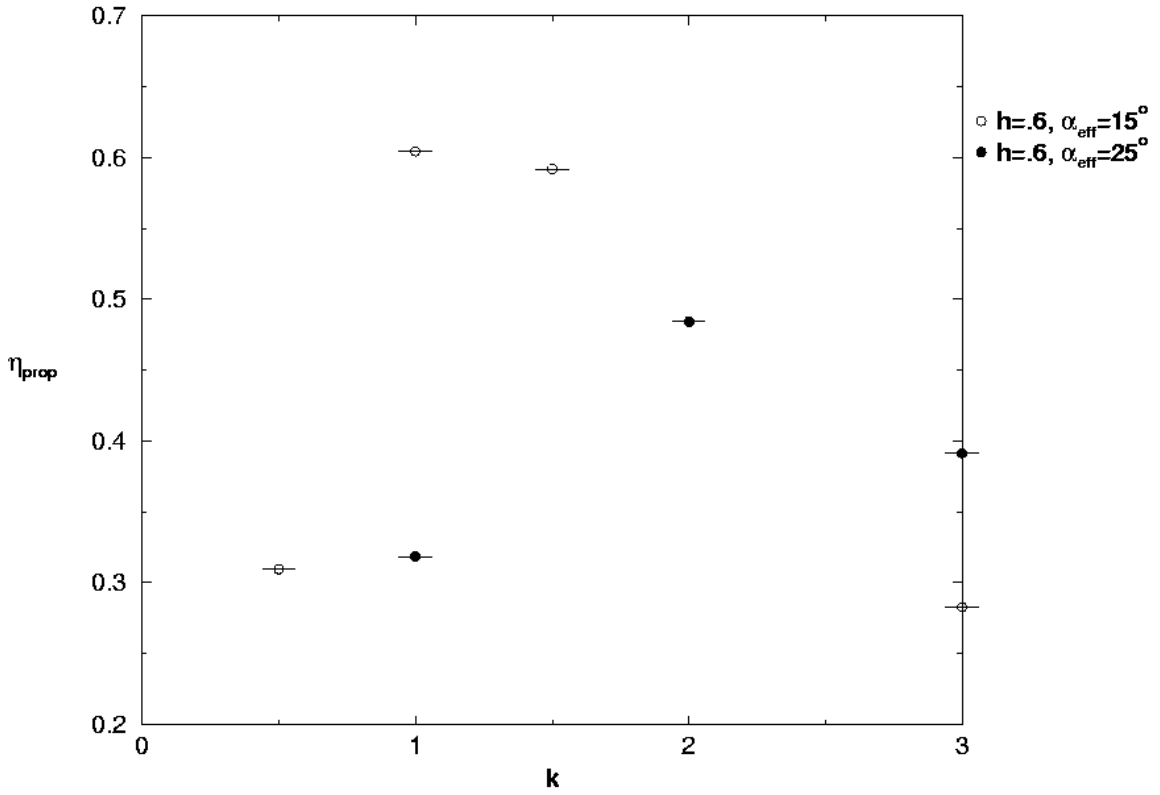


Figure 10. Efficiency versus reduced frequency for h=.6, x_p=.25, ϕ=90°

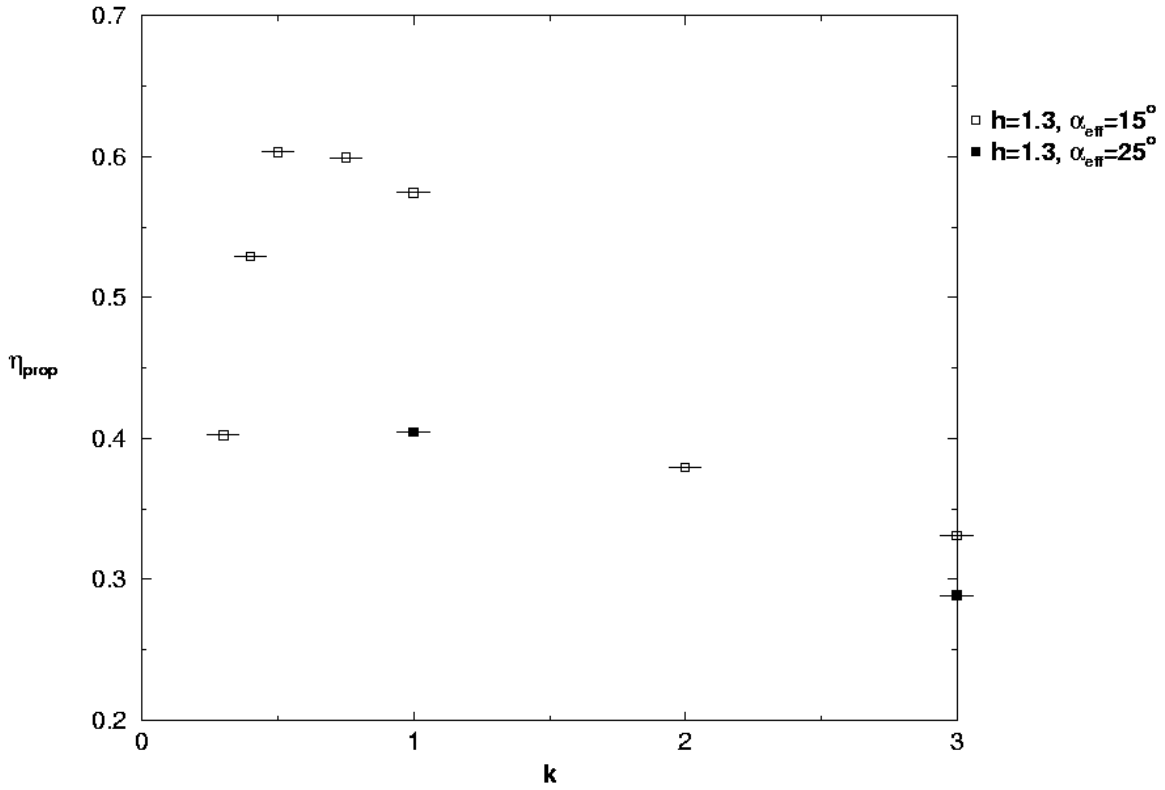


Figure 11. Efficiency versus reduced frequency for h=1.3, x_p=.25, ϕ=90°

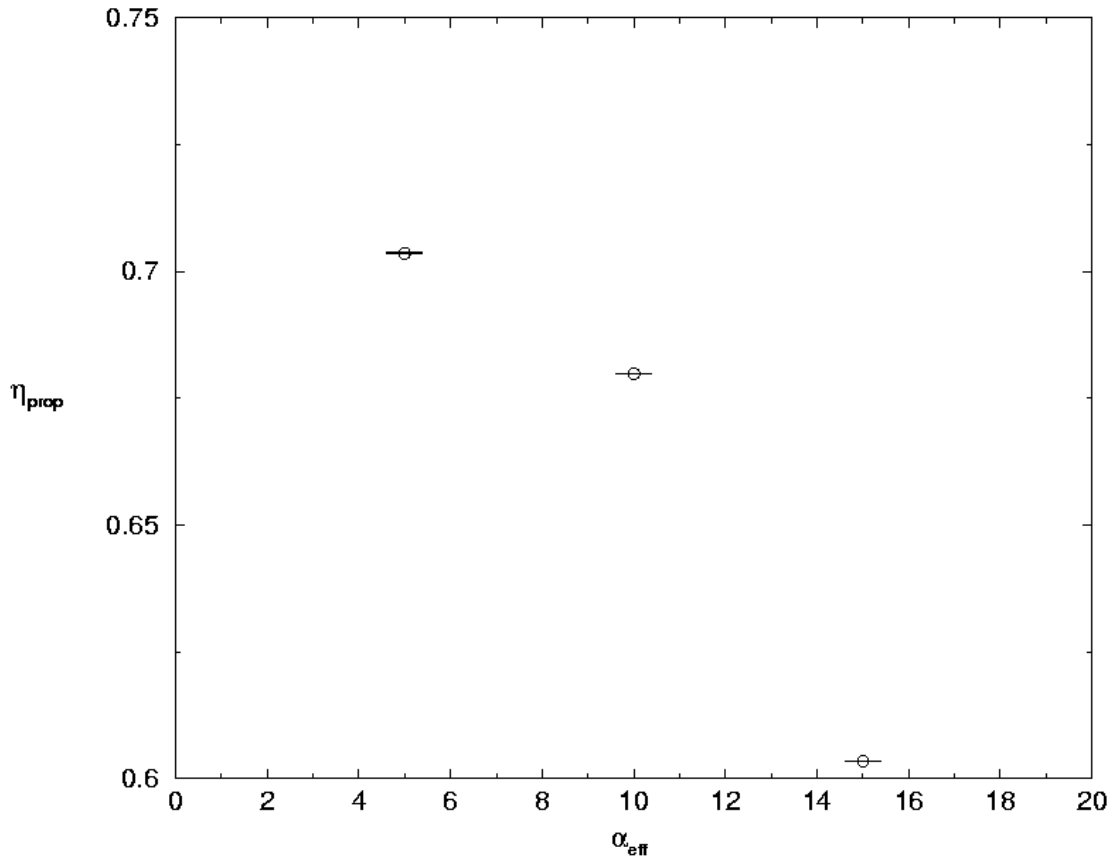


Figure 12. Angle of attack sweep for k=.5, h=1.3, x_p=.25, ϕ=90^o

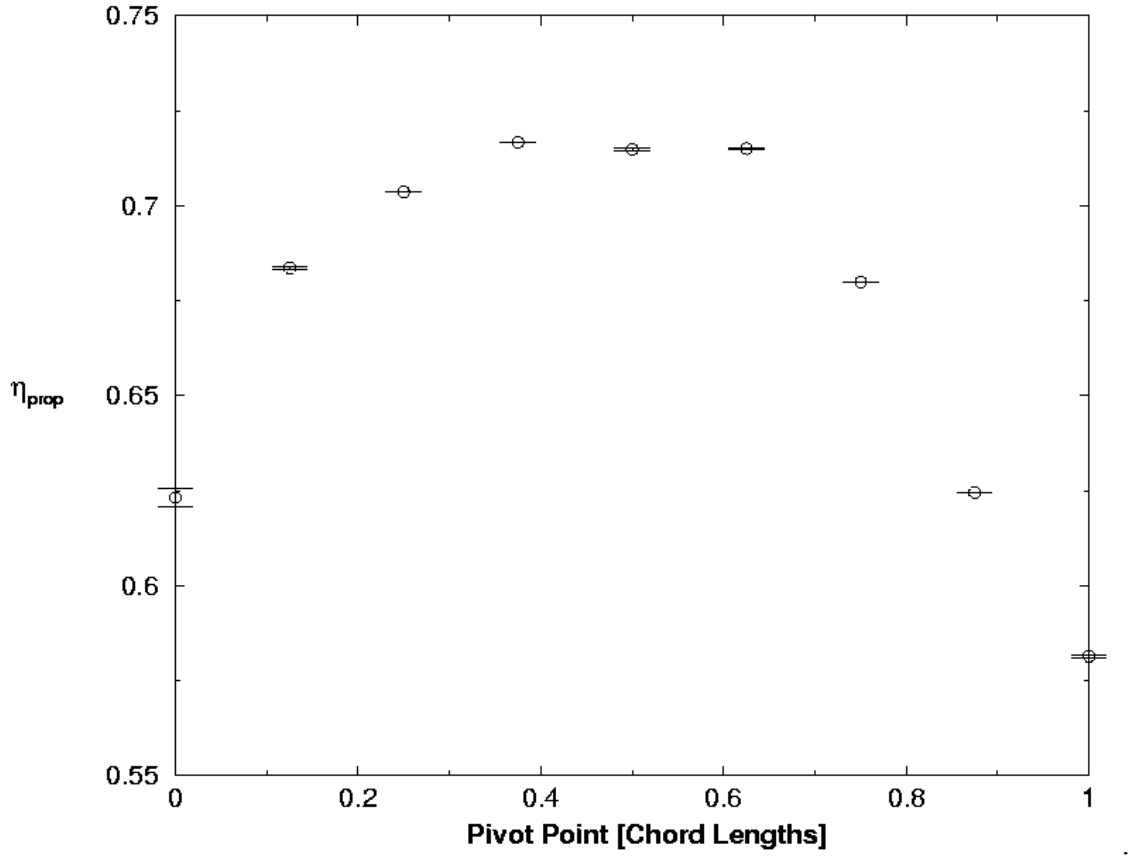


Figure 13. Pivot Location Sweep, h=1.3, k=.5, α_{eff}=5°, ϕ=90°

An additional case was run to look at the possible advantages gained by placing the airfoil ahead of the mast. These results are shown in Figure 14, and clearly indicate that there is no benefit in a forward position of the airfoil.

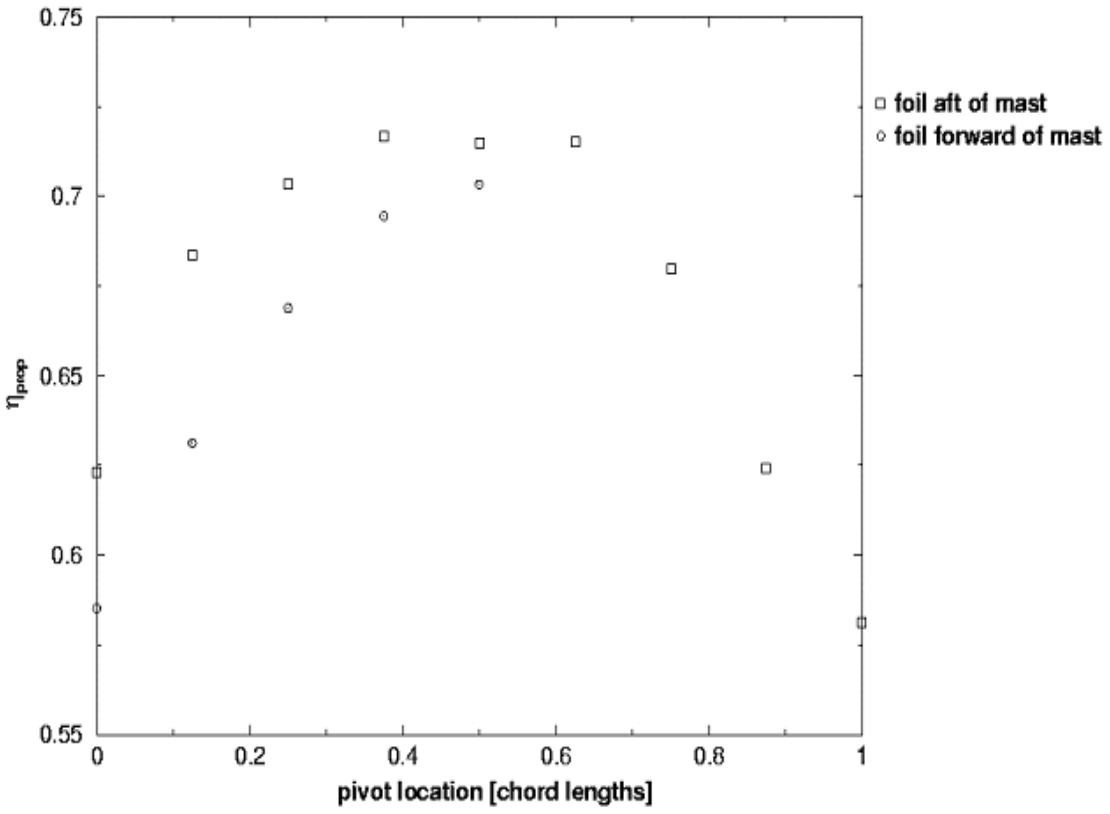


Figure 14. Forward positioning of airfoil, h=1.3, k=.5, α_{eff}=5°, ϕ=90°

Finally, the phase angle between the pitch and plunge motion was varied. As seen in Figure 15, this produces a clear peak in performance of the airfoil at 90°.

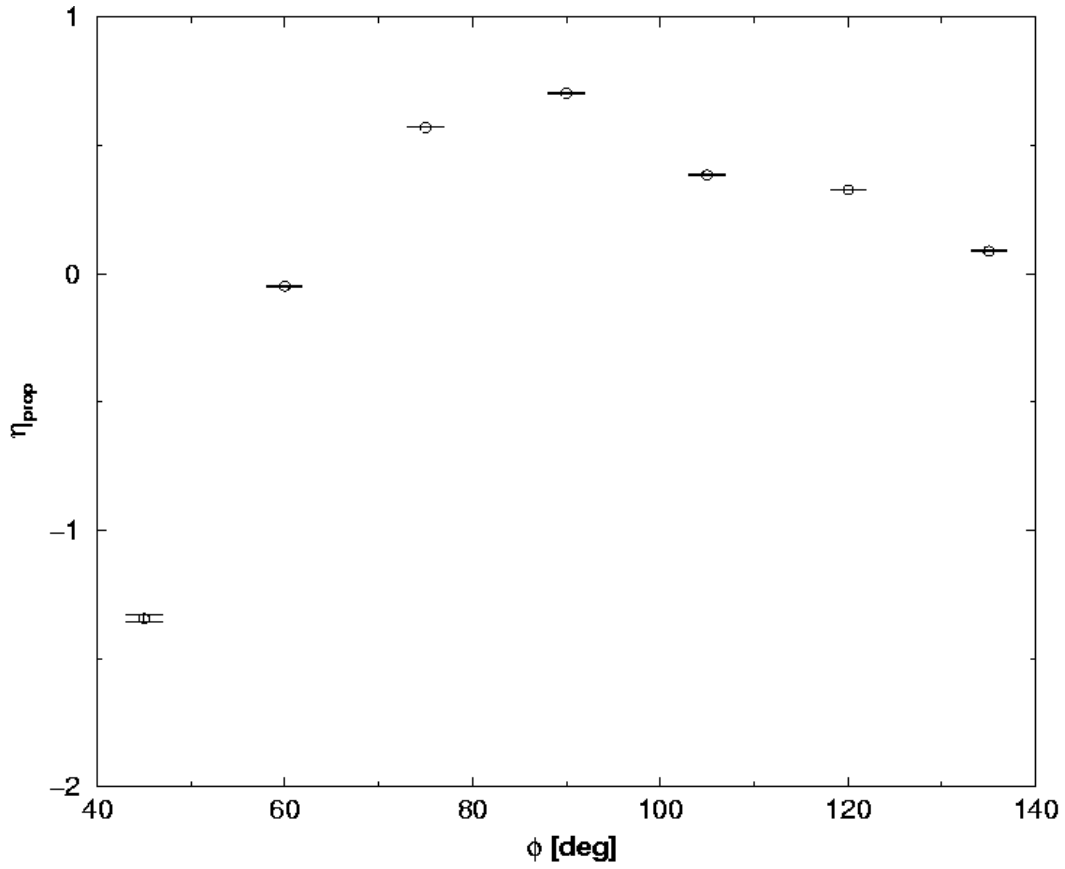


Figure 15. Phase sweep, h=1.3, k=.5, α_{eff}=5°

The numerical optimization method produced a clear position for peak airfoil efficiency. Table 1 summarizes these findings.

Table 1. Airfoil Peak Performance Parameters

k	0.5
h	1.3
α_{eff}	5°
α_{geo}	20.024°
x_p	0.375
ϕ	90°

C. VISUALIZATION

In addition to providing efficiency and thrust predictions, the NS analysis is also used to provide flow visualization simulations. We ran 5 cycles, saving 72 individual frames from the last cycle of each NS simulation. By using the UNIX based Animate program, these images can be used to produce an animation that illustrates the airfoil's

movements and the resulting wake profile. By observing the resulting wake profile, one can gauge the effects of phenomena such as vorticity and dynamic stall. Additionally, the downstream wake profiles can be used to predict the best phasing and angle of attack for the aft foil.

In order to observe the effects of airfoil position, two baseline cases of the airfoil were run. The first case was with the airfoil feathered according to Equation 7, or at a α_{eff} of 0^o, as shown in Figure 16. As can be seen, the airfoil is not operating in a feathered case, producing a $\bar{C}_T = .454$. This is due to the large vortices generated by the rapid pitching of the airfoil through almost 140^o at top and bottom dead center. The large turning angle is caused by the large geometric angle of attack needed to produce a null effective angle of attack. Also of note is that Equation 6 is only true when x_p=.00. In this case, the effective angle of attack will be larger than predicted since x_p=.25.

For a feathered case; it is assumed that k is low and the foil passes through the flow without producing a significant wake. Clearly, this in not true here. For a finite k, it is impossible to produce a true feathered case, however it is possible to minimize the C_T by using small values of k.

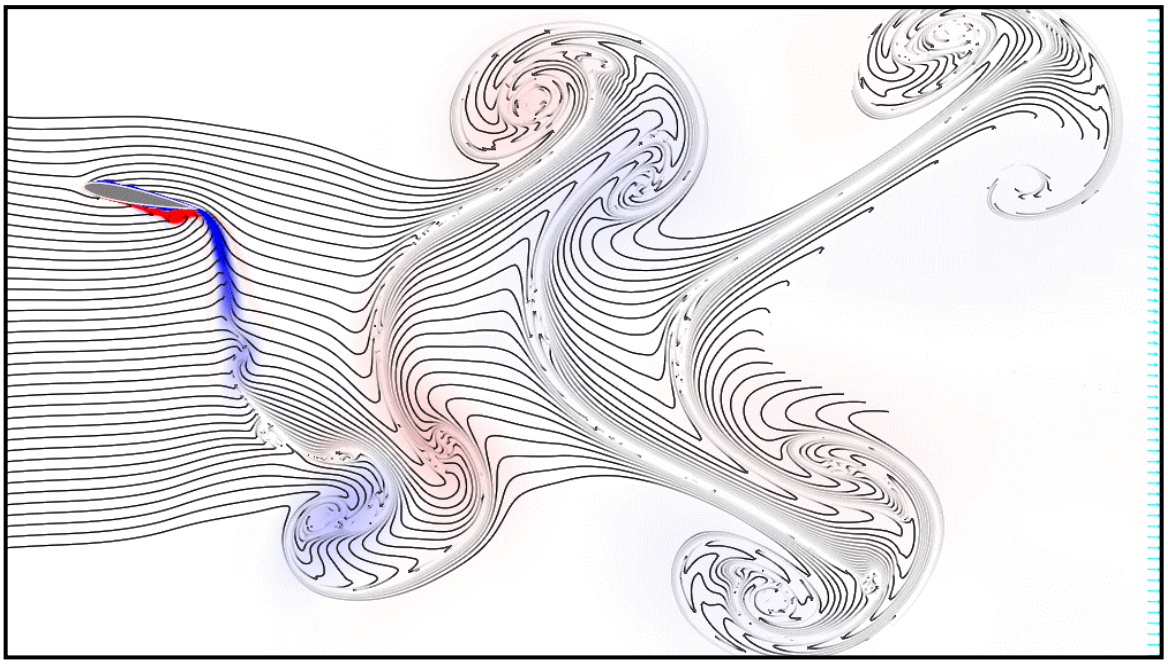


Figure 16. Feathered Case, k=2, h=1.3, x_p=.25, ϕ=90^o

In contrast to the feathered case there is the pure plunge case, as illustrated in Figure 17. Easily visible in the illustration are the leading edge vortices, which result in reduced thrust and diminished efficiency. For this case, C_T=.717, but η_{prop}=7%.

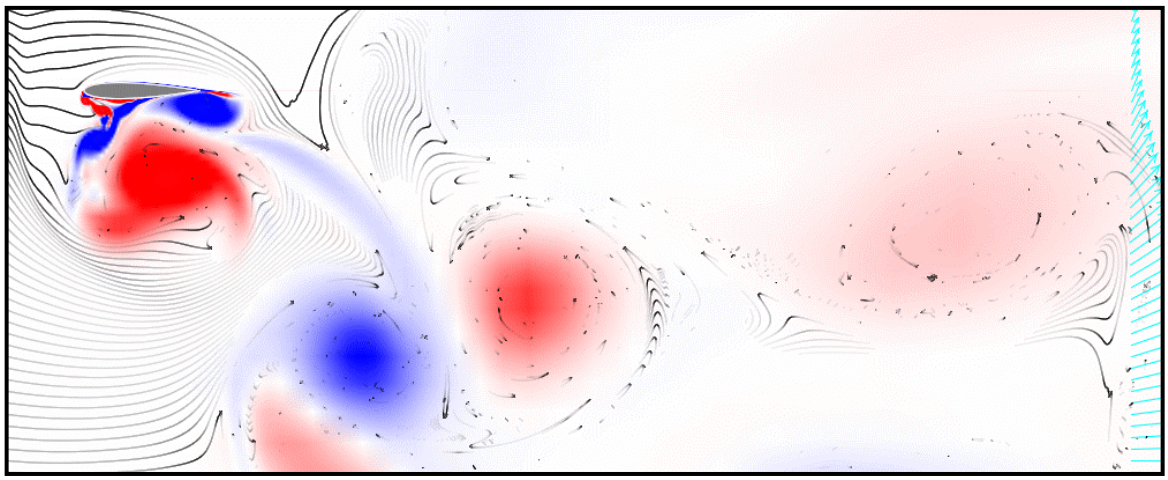


Figure 17. Pure plunge case, k=1, h=1.3, x_p=.25, ϕ=90^o

Of more interest to this study are the visualizations of the peak performance case. The case summarized in Table 1 is illustrated in Figure 18. As can be seen in the visualizations from the downstream velocity indicators, this airfoil is thrust producing. Also of note is the lack of large dynamical stall vortices being produced by the airfoil motion.

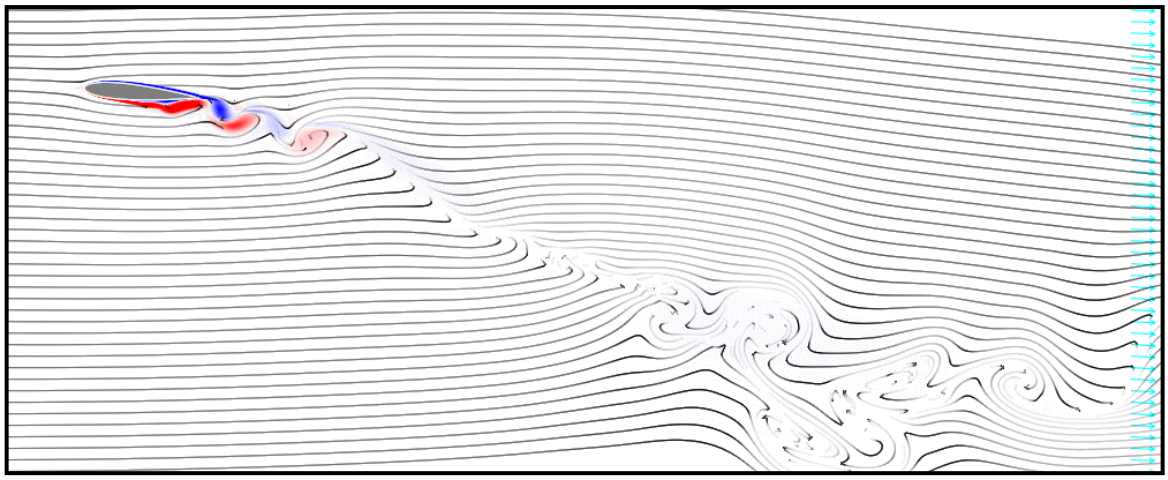


Figure 18. NS peak performance case, k=.5, h=1.3, x_p=.375, ϕ=90^o

D. APPLICATION TO EXPERIMENTAL ANALYSIS

Through use of the visualizations covered in the preceding section, the wavelength of the wake effects can be calculated. From these measurements, the downstream characteristics of the wake can be predicted. Of the most importance is using these characteristics to predict the optimum phasing for the rear wing. Proper positioning of the rear airfoil will allow it to take full advantage of the energy in the wake created by the forward airfoil, thereby increasing the aft foil's performance.

Using the visualization software developed by Dr. Jones, flow visualizations of the downstream velocities were produced using the NS results, such as the visualization seen in Figure 19. The software also output the *u* and *v* components of the velocity, along with the angle of the resultant velocity. A region of flow with higher than free stream velocity would result in an increase of the amount of thrust generated by the aft airfoil. Additionally, if the aft airfoil was timed in such a way that the airfoil was plunging upwards through the region with the largest downward angle of flow, or vice versa, the aft foil thrust would be increased by canting the normal force vector of the aft airfoil further forward. Figure 20 shows an enlarged view of the NS prediction for the flow region about the aft airfoil. The two vertical lines mark free stream velocity, so it can be easily seen where the local velocity exceeds the free stream value.

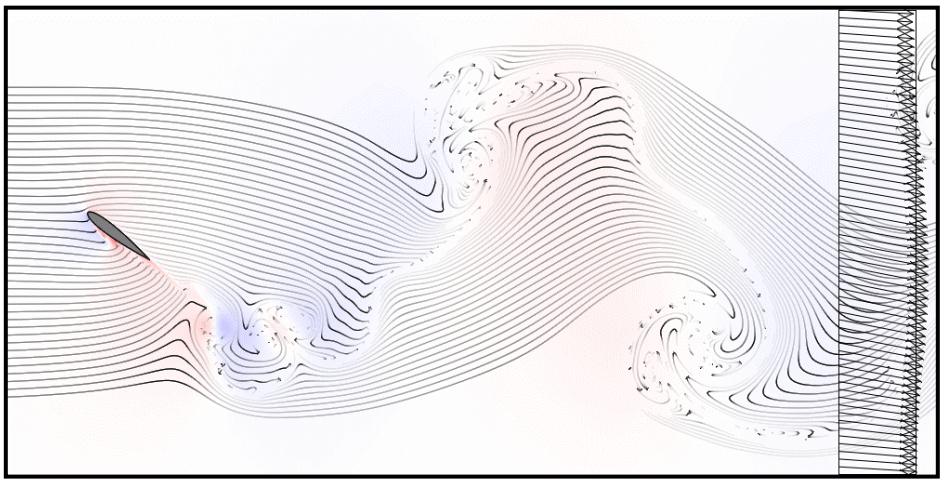


Figure 19. NS flow visualization of downstream velocities

Figure 20. Enlarged view of NS downstream velocities

IV. EXPERIMENTAL SETUP

A. APPARATUS

The experiments were conducted in the Naval Postgraduate School Aeronautics Department's water tunnel. The water tunnel utilized is a closed circuit, continuous flow facility. The test section measures 38 x 51 x 150 cm. Figure 21 shows a schematic of the water tunnel. Flow through the water tunnel can be varied from 0 to .5 m/s. The flow is measured by an impeller type flow meter in the return loop. This flow meter is a low fidelity device and the data it produces is unreliable, therefore, LDV data was used to estimate deviation.

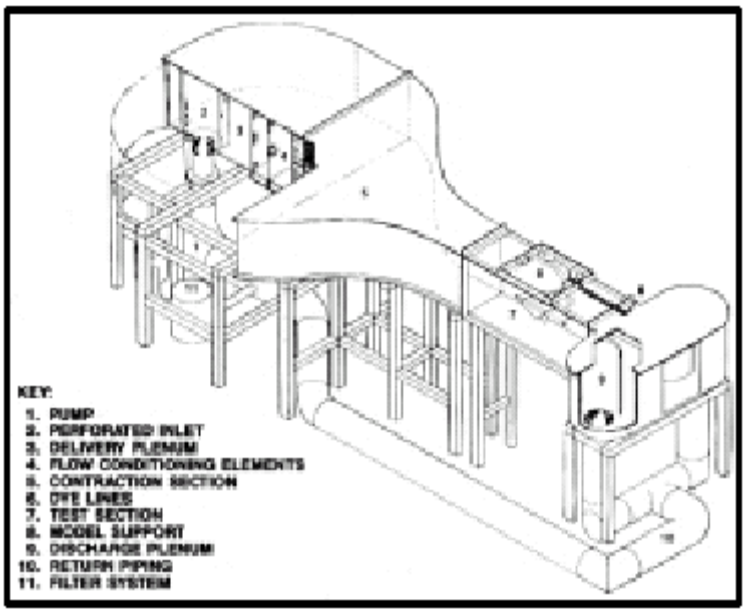


Figure 21. Schematic of NPS Aeronautics Department's Water Tunnel

The experiments were conducted using Dr. Jones' flapping foil device. The device is an electrically driven, dual airfoil device, shown in Figure 22. The electric motor is coupled to the device through a reduction gearbox and drives both airfoils pitch and plunge motions. These motions can be set independently for each foil by altering the position of the connecting rods on both the bell cranks and actuators.

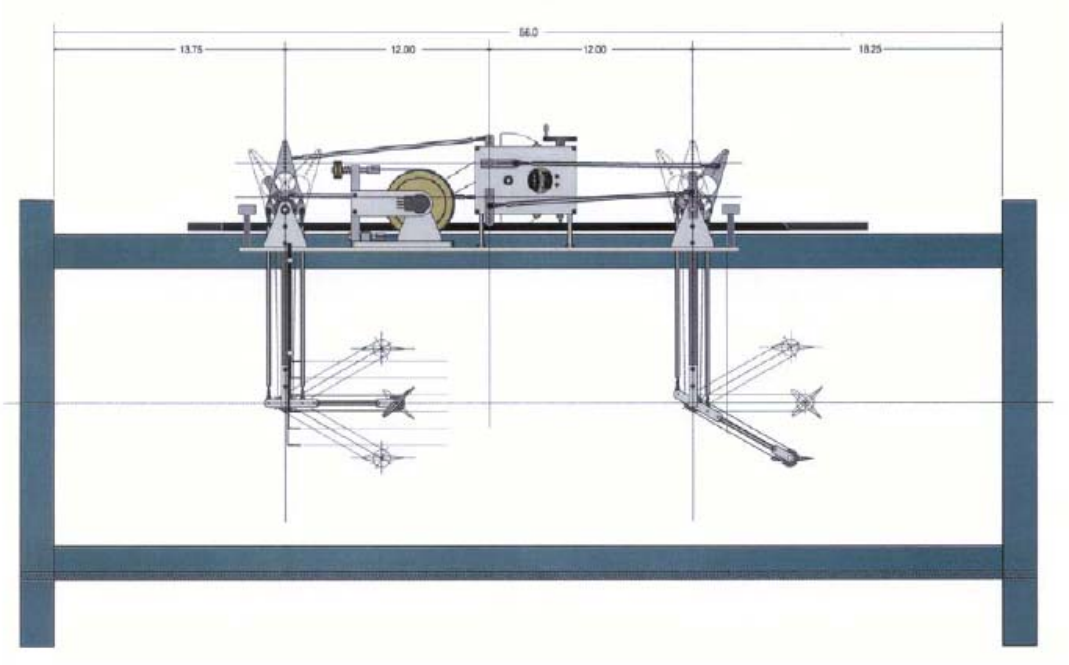


Figure 22. Jones' wingmill modified for thrust production

Figure 23 is a close up of one of the airfoils, describing in detail its motions. Each foil is connected to a strain gauge via the mast. The mast is set up as a lever to transmit the force produced or received by the foil/mast combination to the strain gauge. The dual strain gauges enable thrust and drag measurements to be collected for both foils.

The data is collected from the device using the system shown in Figure 24. The strain gauge signals are sent to the strain meters where the voltages are converted to analog values. The meter then sends these analog values to the DAC in the PC to allow for data collection and conversion back to digital values. The motor frequency is collected by using an optical encoder. The optical encoder output is sent to both the oscilloscope and the DAC. The oscilloscope allows for visual confirmation of the motor's speed. The DAC card input allows the frequency the motor is running at to be recorded. The DAC used is an Omega DAQP-16. An Elenco MX-9300 All-in-one instrument provides the voltage necessary to drive the motor and power the optical encoder.

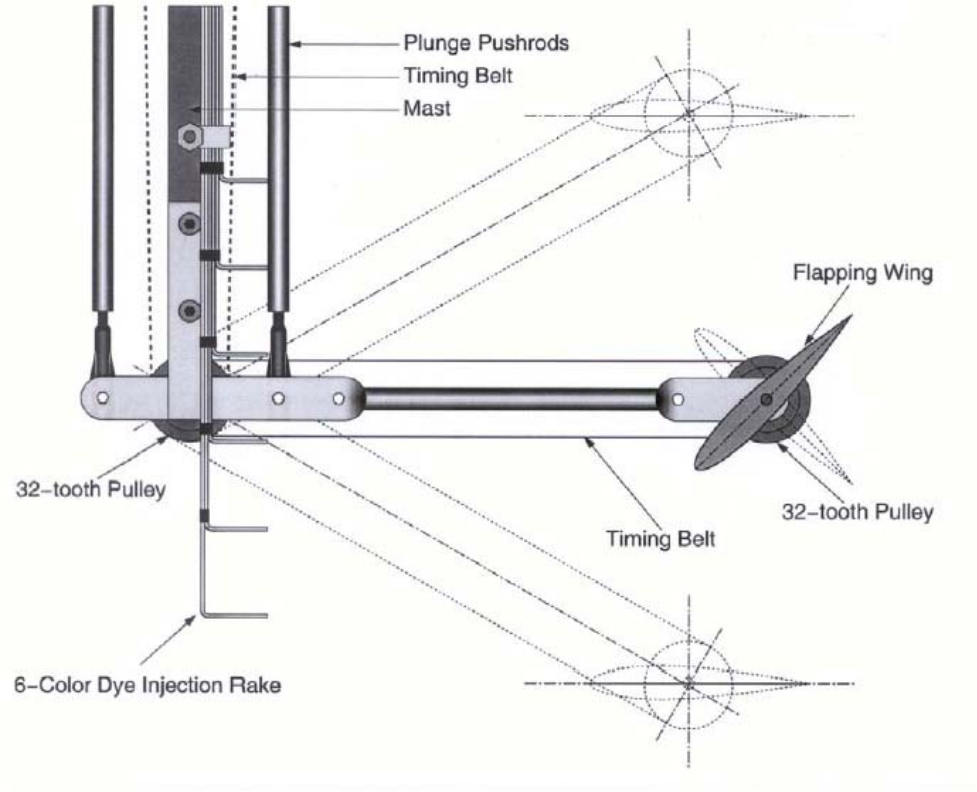


Figure 23. Close up of mast and foil assembly

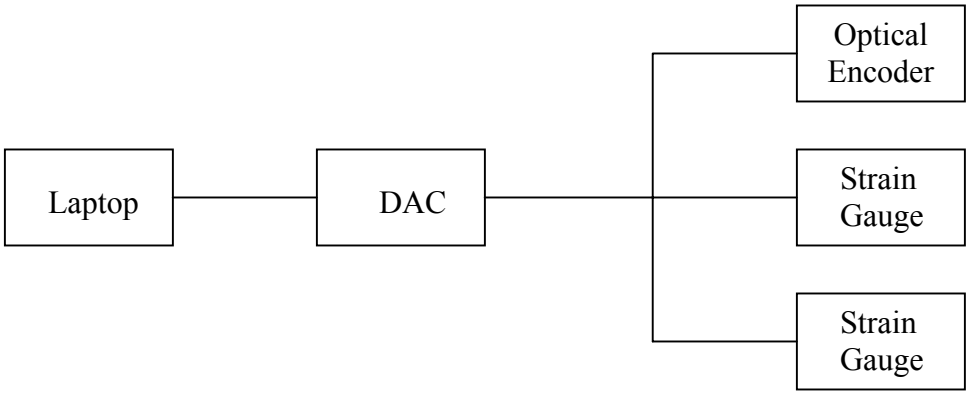


Figure 24. Data collection setup

B. PROCEDURE

1. Calibration

Calibration of the device required a two step process: first, it was necessary to obtain the correlation between the signal received from the strain gauges and the value of this measurement in terms of a measured force, i.e. Newtons. Second, the offset of the force reading from the zero reading had to be determined.

The relationship between the voltage from the strain gauge and the force reading is linear. This linear relationship makes the calibration of the strain gauges a simple process of finding the slope of the line. As each strain gauge reads independently, this calibration was performed twice.

First, a strain gauge reading was taken with no load on the load cell to provide a zero value. Then, known weights of increasing size were hung on the arm of the wingmill. These measurements provided a correlation between a known weight and the voltage from the strain gauge. By plotting these values for each strain gauge, as shown in Figures 25 and 26, using a linear regression fit to the points, a slope and a standard deviation of the slope can be determined.

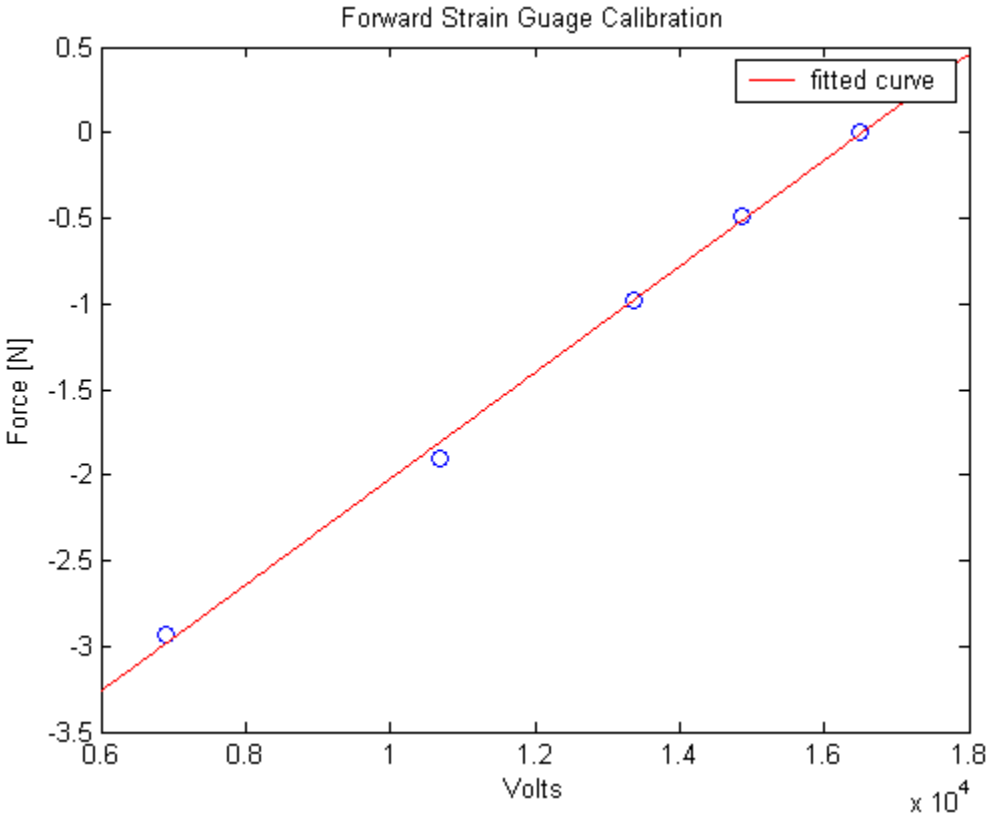


Figure 25. Forward Strain Gauge calibration plot

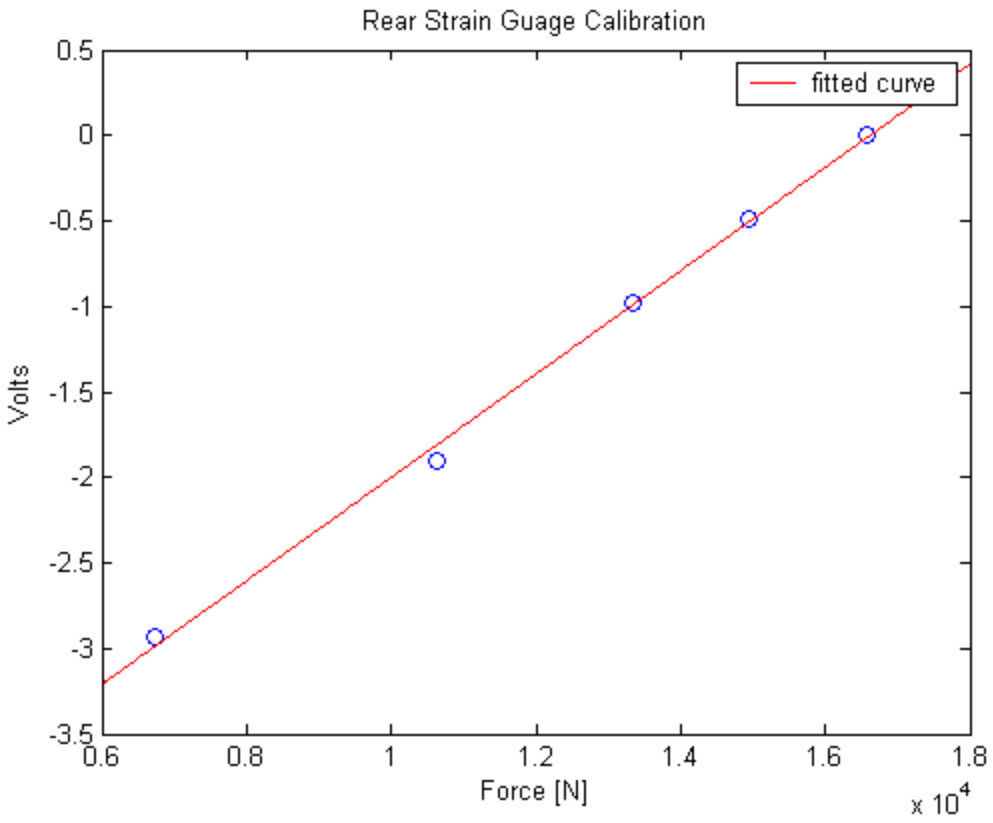


Figure 26. Rear strain gauge calibration plot

The offset of the strain gauge line is caused by several factors: drift, buoyancy, device configuration, and drag of the components below the free surface. Drift contributes the smallest amount to the offset reading. Drag of the components will vary with configuration. Buoyancy of the components was never determined, but determining the zero offset effectively removes it from the force readings.

To obtain the offset value, it was necessary to run the wingmill in its desired configuration, but hopefully without producing thrust. In this mode, the strain gauges will measure the amount of drag caused by the device. Once the data has been obtained, determining thrust is the simple matter of subtracting the offset reading from the measured force reading for each strain gauge.

2. Data Collection

The wingmill was run in numerous different configurations in order to optimize the thrust produced by the device. Similar to the NS procedure, the wingmill was run through several manual optimization procedures.

First, the wingmill was configured as a single foil device to confirm the validity of the NS predictions. In this configuration, the device was run through several variable sweeps to compare experimental data with the numerical data. Variable sweeps included k, α, and h. The variable sweeps served the additional purpose of optimizing the motion of the forward foil.

Following the optimization of the forward foil, the rear foil was installed. The phase angle between the forward and aft foils was predicted from the NS results to be -20^{o}. The first test run was a sweep of the phase angle between the forward and aft foils in order to test the phase angle predictions. Next, the forward foil was fixed and the aft foil was run through an α_g, α_{eff}, and an h sweep to determine its optimum motion.

3. Data Analysis

Each data set was collected on the DAC and stored on the PC. This data was then moved to an SGI workstation for data analysis. A Fortran program, included in Appendix A, was written by Dr. Jones to count the number of cycles the wingmill would run on each data set. The program would then provide time-averaged values of T, C_T, f, k, and their standard deviations for each foil across the full run.

V. ERROR ANALYSIS

A. SOURCES OF ERROR

There are numerous factors that contribute to the error determined in the data analysis section. One ever-present source of error in all measurements was the measurement of flow speed through the test section. To compensate for the lack of high fidelity flow measurement, the pump was run at a continuous speed and the uncertainty measured by Dohring's 1996 LDV survey of the water tunnel test section was adopted (Ref 13). In all cases the motor was run at 20Hz, which should yield a velocity of .203 m/s, and for speeds in this range Dohring measured deviations of about .006 m/s with a turbulence intensity of 3%.

During the calibration process, a linear regression was fit to the voltages corresponding to the known weights, as shown in Figures 25 and 26. From this linear regression there is a standard deviation of the slope.

Each thrust measurement is determined by taking the difference of the loaded force measurement and the feathered force measurement. The feathered measurements are time-averaged values, and therefore contain their own standard deviations, which factor into the errors for each thrust measurement. The thrust calibration error is determined by using Equation 9,

$$\Delta_{cal,thrust} = \left|F_0 - F_T\right| \delta_{cal,foil} \tag{9}$$

where F_0 - F_T is the net force measurement, and δ_{cal} is the calibration error or standard deviation of the slope.

The components of the total error are summed in quadrature as shown in Equation 10,

$$\frac{\Delta_{thrust}}{|T|} = \sqrt{\left(\frac{\sigma_{drag,foil}}{T}\right)^2 + \left(\frac{\sigma_{foil}}{T}\right)^2 + \left(\frac{\Delta_{cal}}{T}\right)^2} \tag{10}$$

which simplifies to the form given by Equation 11.

$$\Delta_{thrust} = \sqrt{\sigma^2_{drag,foil} + \sigma^2_{foil} + \Delta^2_{cal}} \tag{11}$$

where $\sigma_{drag,foil}$ is the standard deviation of the feathered force and σ_{foil} is the standard deviation of the loaded force for the given foil, and the subscript *foil* represents either fore or aft. Figure 27 illustrates the relative contribution of each error source to a measurement of C_T in a typical data set. Note that these errors are summed in quadrature, so the total error is not the direct sum of the values given in the figure.

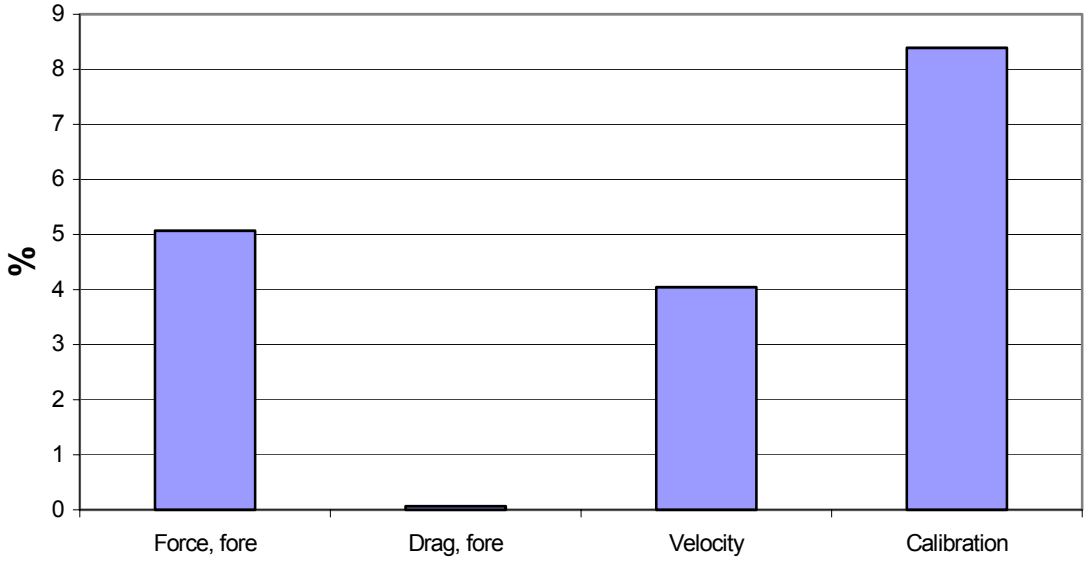


Figure 27. Error sources in a typical data set

The large size of the velocity error is due to how velocity factors into C_T. Since velocity is squared in the C_T calculation, it contributes double to the overall C_T uncertainty.

Error is also present in the computed value of k due to frequency error and velocity error. Since the frequency for each data set is a time averaged value, there is a standard deviation to this measurement. The frequency error is simply the standard deviation of the different frequency values collected.

VI. EXPERIMENTAL ANALYSIS

A. ZERO OFFSET

While the calibration constant for each strain gauge was found in the calibration process, the strain gauges exhibit drift over time. This drift causes a zero offset in the strain gauge calibration that must be accounted for. Also, in the NS code we only integrate the pressure force on the wing. This can result in drag or thrust, but the skin friction is ignored completely. Therefore, for the experimental results to be accurately compared to the numerical results it is necessary to separate the drag of the below free-surface components from the force readings from the strain gauges. To compensate for both of these issues, it was necessary to take a zero offset reading for each data set. Basically, the zero offset is taking a force reading at a point where no thrust is assumed to be produced, or as close to the feathered case as possible.

The evolution of the zero offset procedure was a long and often teething process. During the strain gauge calibration process, it was found that different foil positions resulted in different strain gauge readings. This made it necessary to have a time averaged drag reading that took into account the drag at all foil positions. The original calibration process called for the device to be run at near-zero speed, resulting in a zero offset with minimal thrust and drag in all foil positions.

It was found, however, that it was necessary to take a zero offset set every time the plunge amplitude of the device was changed. Also, in running the device slowly, a huge drag was generated by having the airfoil being almost normal to the flow. This also caused wake disturbances, resulting in poor F_0 measurements. A better F_0 method had to be developed. To remove thrust from the zero offset, the device was run with the foils in a feathered position. This approach would still allow the drag of the underwater components to be calculated, and would also effectively remove thrust from the zero offset values. Also, the zero offset set could be collected with the device running at test speed, thereby reducing the time required for each zero offset.

Rather late in the experimental analysis, it was found that some calibration sets were producing F_0 sets with inconsistent F_0 values and unusually high uncertainties.

Plotting the strain gauge response versus the time revealed that the system was requiring upwards of two minutes to reach equilibrium, as seen in Figure 28.

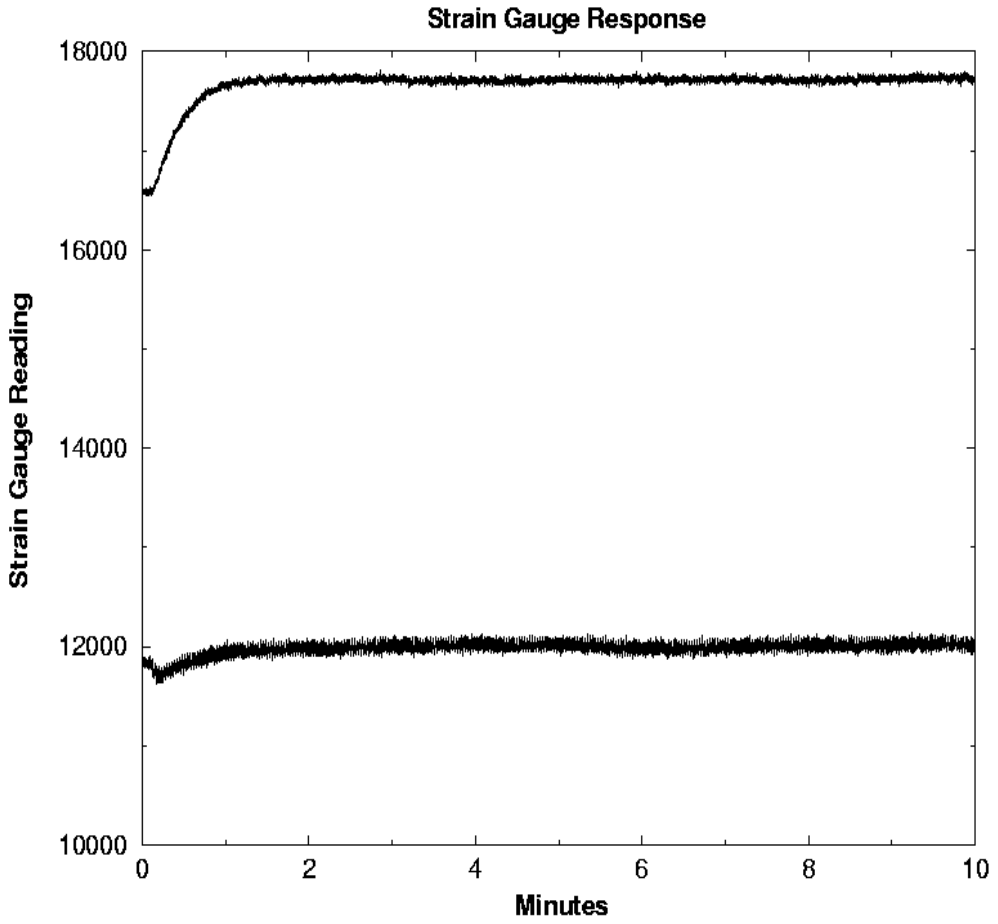


Figure 28. Strain Gauge response for aft (top line) and forward foils.

This issue was simply dealt with by allowing the device to run for several minutes before the zero offset data was collected. This allowed ample time for the strain gauges to begin reading a constant value and provided much more accurate F_0 values.

B. SINGLE FOIL OPTIMIZATION

The first cases looked at experimentally were single foil ones. For these runs, the aft foils were removed from the device to experimentally replicate the NS solutions. The device was then run through several parameter sweeps in an attempt to optimize the performance. First, the device was set to h=1.3, equal to that used in NS. The geometric

angle of attack for the airfoil was then varied, which resulted in a changing effective angle of attack. Figure 29 illustrates these results, with C_T peaking at an effective angle of attack of 15°, and it remains peaked over the range of 10° to 20°. Smaller geometric angles of attack resulted in thrust production and therefore positive values of C_T. Larger geometric angles of attack result in power extraction, or negative C_T values. The cases that approach a C_T of zero are those that are theoretically feathered.

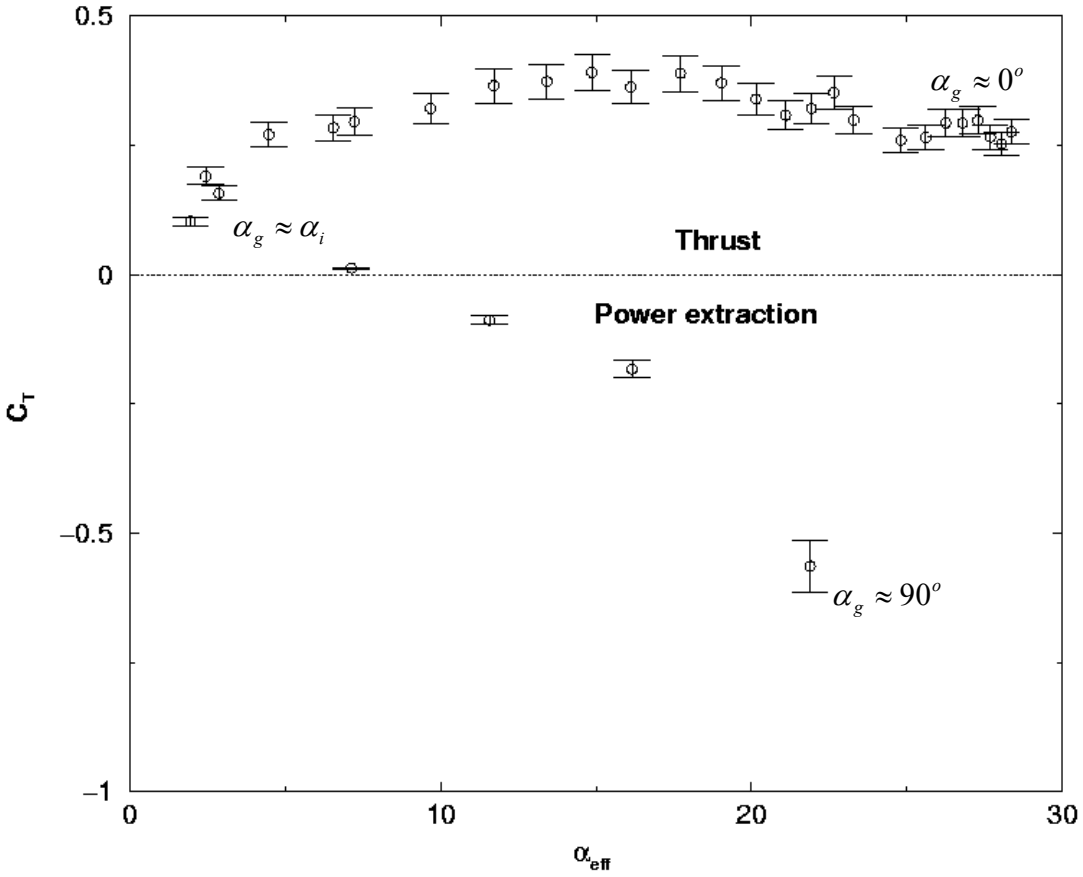


Figure 29. Single foil effective angle of attack sweep, h=1.3, k=.5, x_p=0, ϕ=90°

The other sweep that was performed was plunge amplitude, illustrated in Figure 30. Increasing plunge amplitude resulted in an increasing C_T. The C_T peaked at h=1.2, and then abruptly declined at higher plunge amplitudes. The reason for the abrupt drop off is unknown, but should be investigated in further studies.

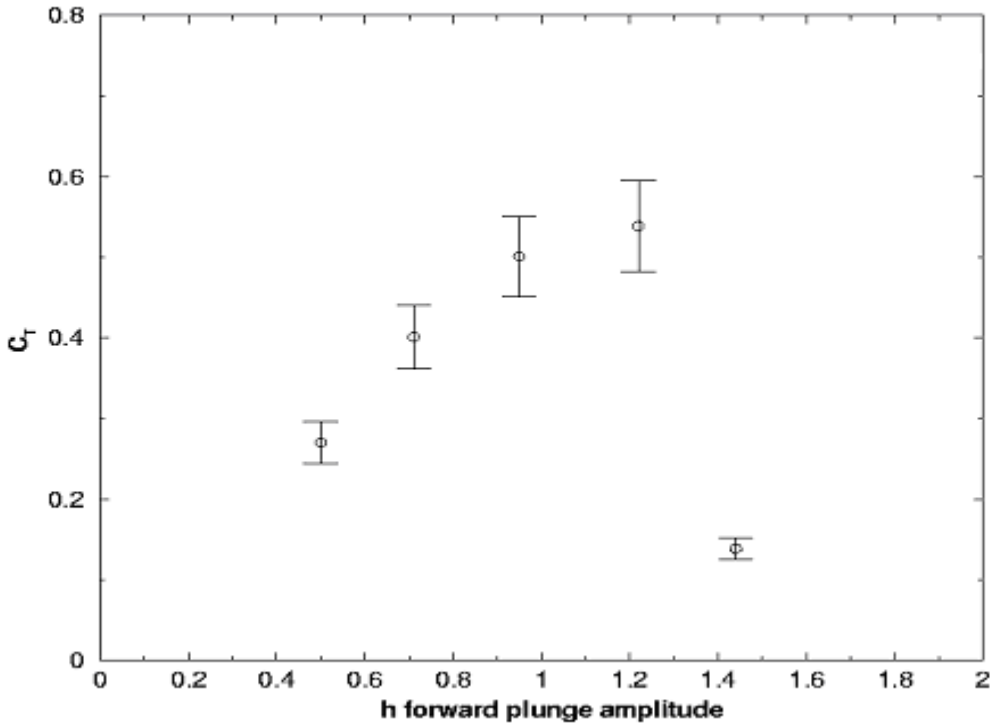


Figure 30. Single foil plunge amplitude sweep, α_{eff}=15^o, k=1

C. NS – EXPERIMENTAL COMPARISON

Following the single foil optimization, the single foil experimental results were compared to the NS results for a similar case. The resulting plot is illustrated in Figure 31.

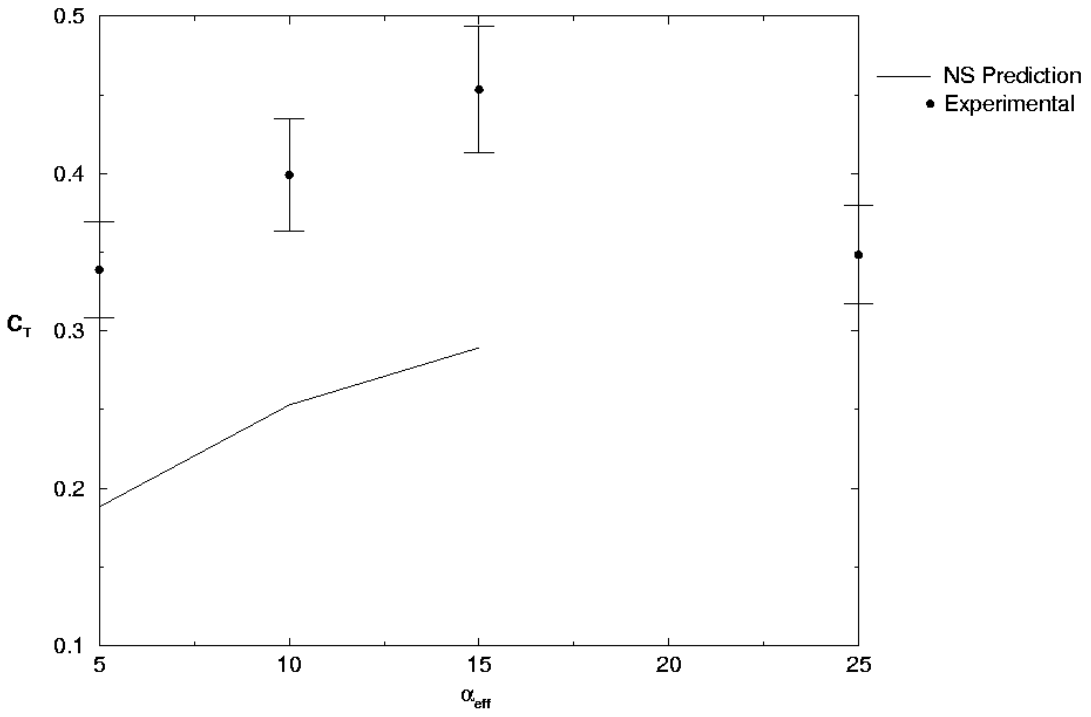


Figure 31. Comparison of NS and experimental results, h=1.3, k=1, x_p=.25

While NS optimization was done with regard to efficiency, all experimental optimization had to be done with respect to C_T. This is due to the lack of an adequate method to accurately measure an experimental power in. As the plot shows, the two methods are in good qualitative agreement, providing a good idea of the trend of the experimental data. However, there is a large discrepancy in the value of C_T recorded experimentally between that predicted by NS. One partial explanation for this is that the experimental data presented here was taken before the final zero offset method had been fully developed, so the trends in the data are correct, but there could be a significant zero-offset error.

D. DUAL FOIL OPTIMIZATION

Prior to the introduction of the aft foil, it was necessary to reduce the number of parameters that could be varied on the device to limit the scope of the research. To simplify the angle of attack calculations it was decided to fix the pivot point about the leading edge. This served to decouple pitch and plunge with respect to effective angle of attack, allowing them to be varied independently of each other. Also, it was decided to fix the phase angle between pitch and plunge at 90^o. NS predictions, seen in Figure 13, indicated that there would be little benefit in altering this phase angle. With the pitch/plunge phase angle fixed at 90^o all further mention of phase angle will indicate the angle between the forward and aft foil motions.

The two goals of the dual foil optimization were to find an optimum thrust from the aft foil and to look for an aft foil thrust value that was above the forward foil thrust. For this to happen, it was necessary to know how the flow would behave in the aft foil region. Initial data for the dual foil device showed a large discrepancy in the performance being generated by the forward and aft airfoils, despite their identical configurations. It was theorized that one possible cause for this was that the flow being seen by the aft foil was faster than free stream. Since the forward foil was thrust producing, it was mandatory that the flow behind the front foil was accelerated beyond the free stream velocity.

To predict the increase in velocity, actuator disc theory, as seen in Equation 12, was used

$$v_{1c} = -\frac{V_c}{2} + \sqrt{\left(\frac{V_c}{2}\right)^2 + \frac{T}{2\rho A}} \tag{12}$$

where v_{1c} is the incremental velocity at the disc, V_c is the free stream velocity, T is the thrust produced by the foil, and A is the area normal to the flow that is swept by the foil [Ref 16]. This results in a v_{1c}, the velocity at the disc, of .233m/s.

This increase in flow velocity would result in an almost 15% decrease in the k of the aft foil. The decrease in k would change the aft foil's induced angle of attack, and

therefore the effective angle of attack, which would explain the loss of performance in the aft foil.

The actuator disc prediction was reinforced through the use of NS predictions. By using flow visualization for a similar case, the flow conditions downstream were predicted. The downstream flow for 8 different foil positions was analyzed. A post-processing program was used to provide downstream velocities. An average of the peak downstream velocities was used to predict the downstream *k*. The *k* predicted by NS was more conservative than the actuator disc prediction, but was still 10% below the free forward foil value.

Using these predictions, a sweep of the phase angle between the forward and aft foils was run. Figure 32 shows the results. Clearly visible is one wide peak in the region of -20^{o} to -40^{o}, which is in agreement with the predicted phase angle of -20^{o}.

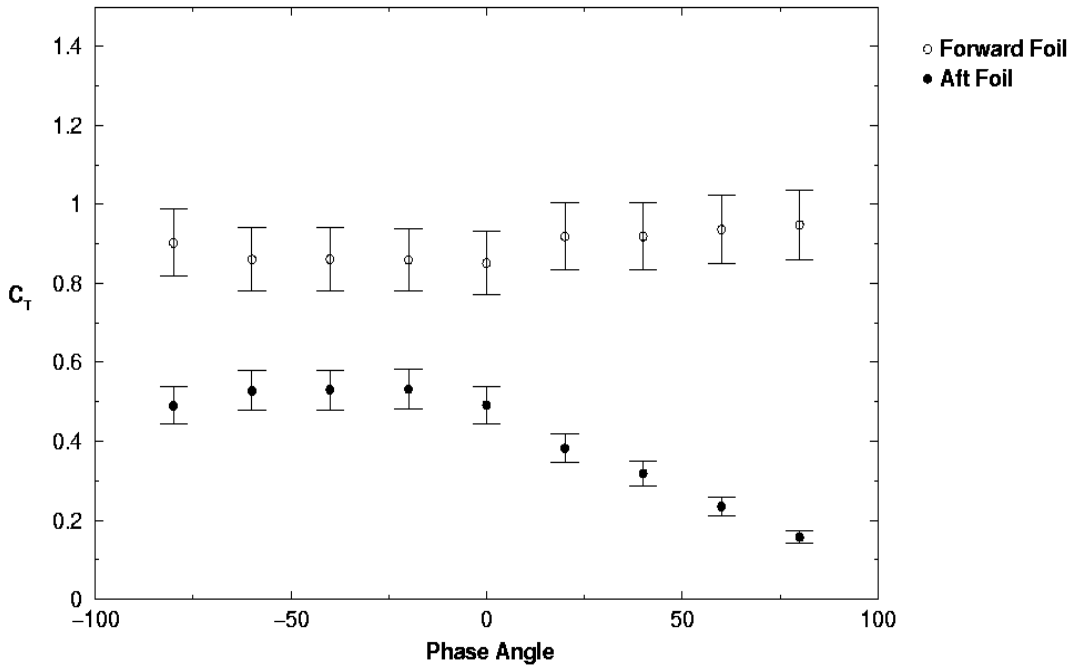


Figure 32. Dual foil phase sweep

In order to peak the aft foil as much as possible, it was necessary to find the ideal geometric angle of attack for the aft foil. While it was known that the aft foil was seeing a different flow, it was unknown how accurate the NS predictions were for the angle of the flow. To measure the angle of the downstream flow, a rudimentary flow indicator, or

windsock, was devised. The windsock was composed of a balsa mast with 9 equally spaced indicators. The windsock was placed in the test section just upstream of the aft foil. The event was recorded with a digital video to allow for image captures. The indicators' steady state position were influenced by their buoyancy, which gave each indicator a different steady state position, as can be seen in Figure 33.

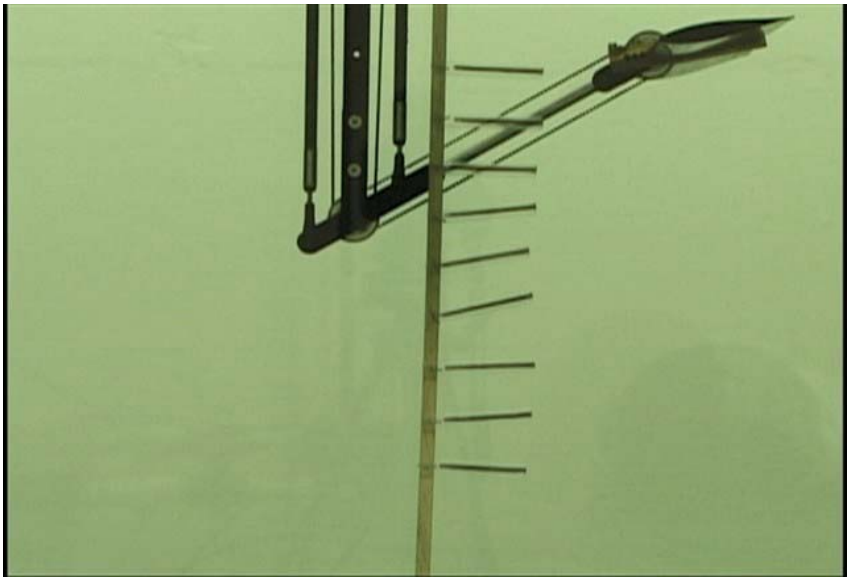


Figure 33. Windsock setup and steady state

The device was then powered up, and positions of the windsock were observed. The resulting images for a cycle of the device are illustrated in Figure 34, with foil positions indicating the forward foil position. The NS flow visualization for similar positions are shown in Figure 35.

From the steady state image, the angle bias of each indicator was measured and averaged to give a value for the mean value of the bias. The bias was then applied to the angle measured for each indicator. The windsock produced angle peaks that are similar to that predicted by the NS solver, thereby validating the downstream NS predictions. Both the NS predictions and the windsock produces estimates for the phase angle that are in agreement with the values predicted by the experimental phase sweep.

With the proper timing of the aft foil known, the geometric angle of attack was swept to maximize the rear foil C_T. Figure 36 illustrates the results for the geometric angle of attack. The peaks correspond to an effective angle of attack of almost 30^o. The

geometric angle of attack that produces the highest C_T agrees with the angle of attack predicted by NS when taking into account the reduced k and the angle of the downstream flow.

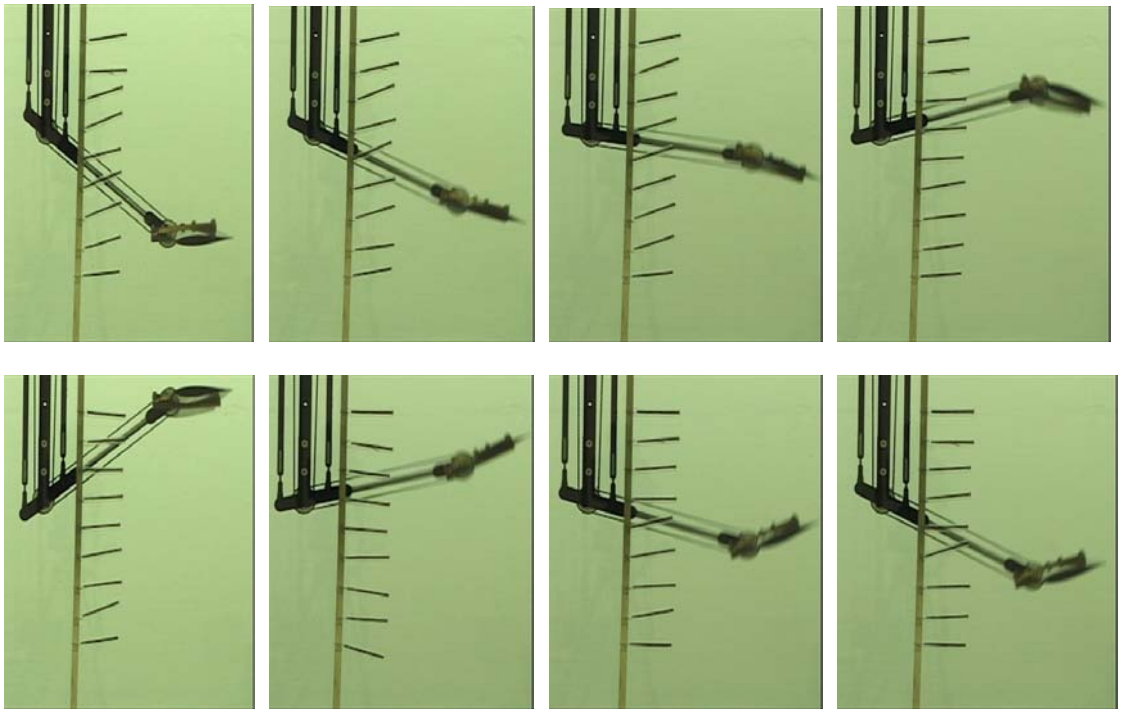


Figure 34. A complete device cycle with the windsock, beginning with bottom dead center in the top left, and going clockwise

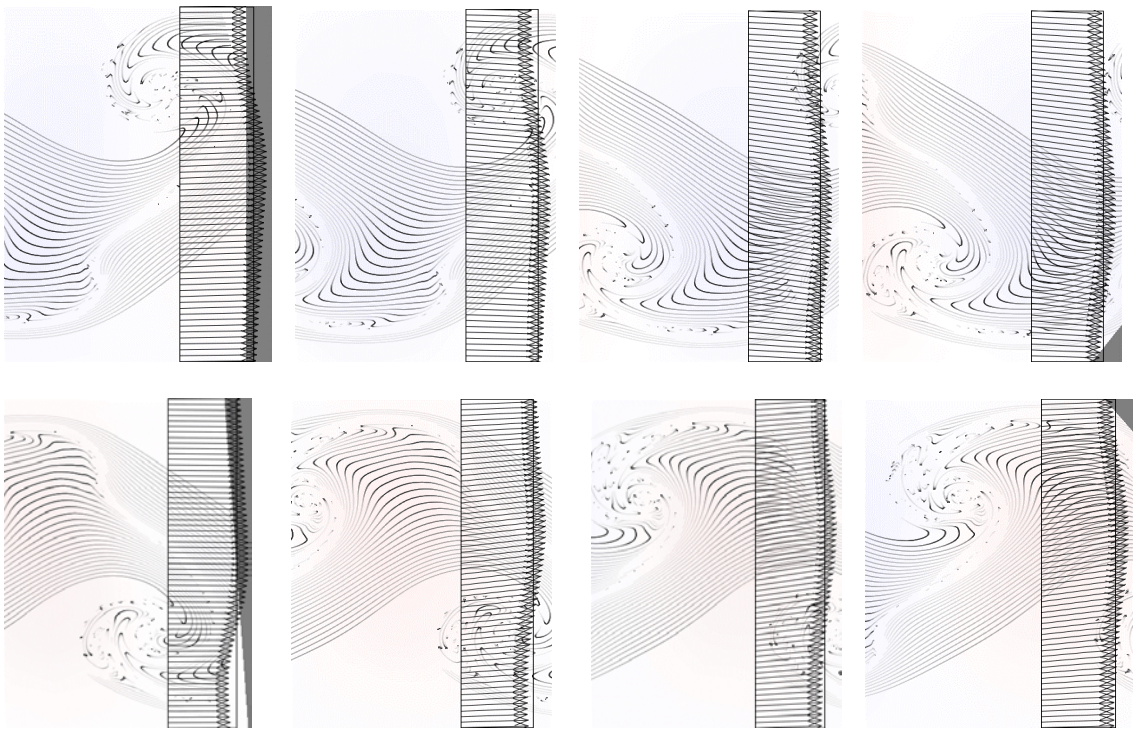


Figure 35. NS predictions of a complete device cycle, beginning with bottom dead center in the top left, and going clockwise

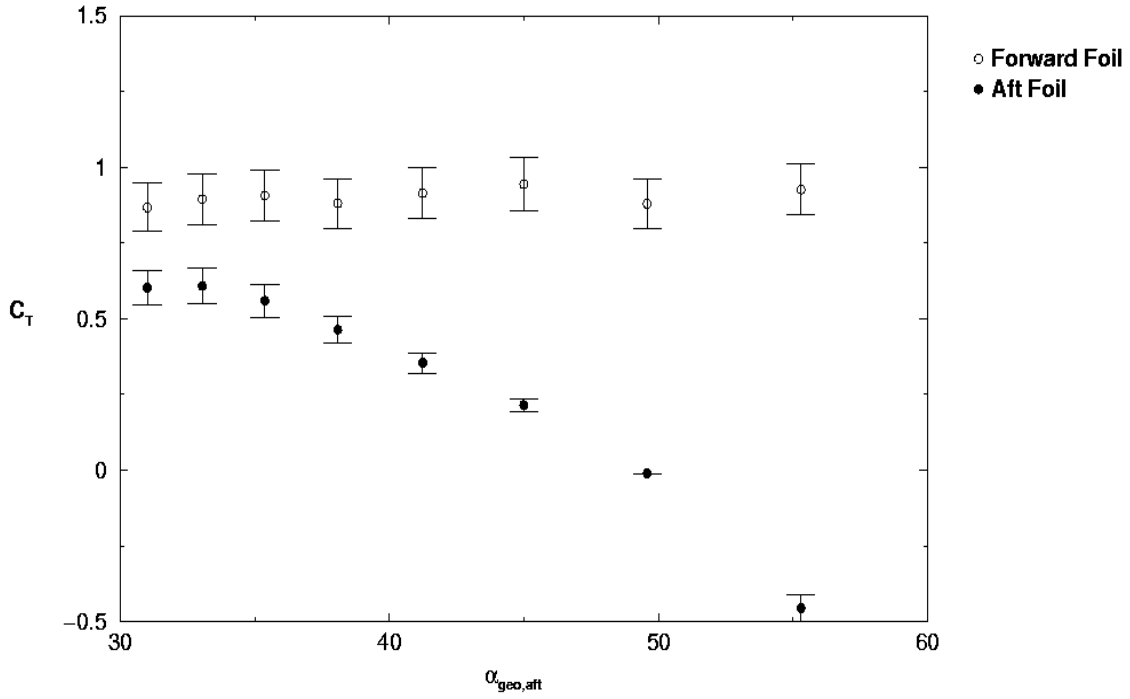


Figure 36. Aft foil geometric angle of attack sweep, h=1.44, k_{fore}=1, ϕ=-20°, $\alpha_{eff,fore}$=15°, $\alpha_{eff,rear}$=30°

VII. CONCLUSIONS AND RECOMMENDATIONS

The NS solutions for a single oscillating foil were a good prediction tool for the experimental analysis. The NS downstream predictions were accurate. The downstream NS predictions were incredibly helpful and used to guide the experimental optimization of the dual foil device. The optimization of the single foil was a seemingly simple task, the optimization of the aft foil, however, left something to be desired. Even with the NS predictions, the aft foil's thrust never exceeded that produced by the forward foil. The aft foil can complement the thrust produced by the forward foil, but it has yet to exceed the forward foil through the use of beneficial wake effects.

Further investigation into the device might yet reveal aft foil thrust that exceeds the forward foil. Future exploration of variables that were ignored in this research could further increase the aft foil performance. Also, this was primarily a two dimensional study of the device. A three dimensional study of the device might reveal unknown benefits in the form of tip vortices or other phenomena.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. FORTRAN CODE

Authored by Jones

MAIN.F

```
       program compute_power
c
       common /dacpar/ cal_rate, thrust_rate, iv_l, iv_h
       common /calibr/ force_slope_fore, force_dev_fore,
     &                 force_slope_aft, force_dev_aft
       common /drags / drag_fore, drag_aft, dragf_dev, draga_dev
       common /thrust/ thrust_fore, thrust_aft, thrustf_dev,
thrusta_dev
       common /freqs / freq_avg, freq_dev
       common /flowpr/ velocity, d_velocity
       common /params/ verbose
       common /filenm/ drag_file, thrust_file, vel_file
       character*30 drag_file, thrust_file, vel_file
c
       character*30 argi
       logical its_there, verbose
c
c----  Set threshold values.
c
       iv_l = 8000
       iv_h = 13000
c
       pi = acos(-1.0)
       chord = 0.0622
       span = 0.1651
       area = chord * span * 2.0
       rho = 999.0
c
c----  Define force calibration slopes and deviations
c
       force_slope_fore = 3.101e-4
       force_dev_fore   = 2.590e-5
c
       force_slope_aft  = 3.021e-4
       force_dev_aft    = 2.600e-5
c
c----  Get command-line args.
c
       verbose = .true.
c      call getarg(1,argi)
c      if ( argi .eq. "" ) call usage()
c      if ( argi(1:2) .eq. "-v" ) then
c      else
c        verbose = .false.
c      endif
c      if ( .not. verbose ) then
c        write(*,1010)
c        write(*,1011)
c        write(*,1012)
c        write(*,1013)
```

```
c      endif
c
       call getarg(1,argi)
       if ( argi(1:2) .eq. "-h" ) call usage()
       if ( argi .eq. "" ) call usage()
       read(argi,*) cal_rate
c
       call getarg(2,argi)
       if ( argi .eq. "" ) call usage()
       read(argi,*) thrust_rate
c
       call getarg(3,vel_file)
       if ( vel_file .eq. "" ) call usage()
c
       call getarg(4,drag_file)
       if ( drag_file .eq. "" ) call usage()
c
       call getarg(5,thrust_file)
       if ( thrust_file .eq. "" ) call usage()

c
c----  Load velocity data.
c
c      call load_velocity_data()
       velocity = .203
       d_velocity = .006
c
c----  Compute drag.
c
       call compute_drag()
c
c----  Compute thrust.
c
       call compute_thrust()
c
c----  Compute total errors.
c
       cal_fore_error = thrust_fore * force_dev_fore / force_slope_fore
       cal_aft_error  = thrust_aft  * force_dev_aft  / force_slope_aft
c
       dcal = cal_fore_error/thrust_fore
       ddrag = dragf_dev/drag_fore
       dthrust = thrustf_dev/(thrust_fore+drag_fore)
c
       thrust_fore_error = thrust_fore
     &                   * sqrt(dcal**2 + ddrag**2 + dthrust**2 )
c
       dcal = cal_aft_error/thrust_aft
       ddrag = draga_dev/drag_aft
       dthrust = thrusta_dev/(thrust_aft+drag_aft)
c
       thrust_aft_error = thrust_aft
     &                   * sqrt(dcal**2 + ddrag**2 + dthrust**2 )
c
c----  Compute nondimensional values.
c
       red_freq = 2 * pi * freq_avg * chord / velocity
```

```
      red_freq_dev = red_freq * sqrt((freq_dev/freq_avg)**2
     &                              +(d_velocity/velocity)**2)
c
      qinf = rho * velocity**2 / 2
c
      ctf = thrust_fore / qinf / area
      ctf_dev = ctf * sqrt((thrust_fore_error/thrust_fore)**2
     &                   +2*(d_velocity/velocity)**2)
c
      cta = thrust_aft / qinf / area
      cta_dev = cta * sqrt((thrust_aft_error/thrust_aft)**2
     &                    +2*(d_velocity/velocity)**2)
c
      write(*,1000)
      write(*,1001)
      write(*,1002) freq_avg, thrust_fore, thrust_aft,
     &              freq_dev, thrust_fore_error, thrust_aft_error
c
      write(*,*)
      write(*,1003)
      write(*,1004)
      write(*,1005) red_freq, ctf, cta, red_freq_dev, ctf_dev, cta_dev
c
 1000 format('#      F      Tf      Ta      dF     dTf     dTa')
 1001 format('#-------------------------------------------------')
 1002 format(6f8.4)
c
 1003 format('#      k     CTf     CTa      dk    dCTf    dCTa')
 1004 format('#-------------------------------------------------')
 1005 format(6f8.4)
c
      stop
      end
```

LOAD_VELOCITY_DATA.F

```
      subroutine load_velocity_data()
c
      common /params/ verbose
      common /flowpr/ velocity, d_velocity
      common /filenm/ drag_file, thrust_file, vel_file
      character*30 drag_file, thrust_file, vel_file
c
      real v(100)
c
      logical its_there, verbose
c
c----  Get the DSO settings.
c
      inquire(file=vel_file,exist=its_there)
      if ( .not. its_there ) stop 'velocity file not found'
      open(9,file=vel_file,form='formatted',status='old')
      call strip_comment(9)
c
      n = 1
      sum = 0.0
 10   continue
```

```
        read(9,*,end=20) v(n)
        v(n) = v(n) * 0.0254
        sum = sum + v(n)
        n = n + 1
        goto 10
 20     continue
        nmax = n - 1
        velocity = sum / nmax
c
        sum = 0.0
        do n = 1, nmax
          sum = sum + (velocity-v(n))**2
        enddo
        d_velocity = sqrt(sum/(nmax-1))
c
        close(9)
c
c----   If verbose, report settings.
c
        if ( verbose ) then
          write(*,1000)
          write(*,1001)
          write(*,1002) velocity, d_velocity
          write(*,*)
        endif
c
 1000   format('_____________________________________________________')
 1001   format('Velocity data:')
 1002   format('   Velocity = ',f6.4,' +/- ',f6.4,' m/s')
c
        return
        end
```

COMPUTE_DRAG.F

```
        subroutine compute_drag()
c
        common /dacpar/ cal_rate, thrust_rate, iv_l, iv_h
        common /calibr/ force_slope_fore, force_dev_fore,
     &                  force_slope_aft, force_dev_aft
        common /drags / drag_fore, drag_aft, dragf_dev, draga_dev
        common /params/ verbose
        common /filenm/ drag_file, thrust_file, vel_file
        character*30 drag_file, thrust_file, vel_file
c
        real freq(100)
        integer iv(500000,3), iavg2(100), iavg3(100)
        character*10 filename
        logical its_there, initialized, verbose
c
        open(10,file=drag_file,form='formatted',status='old')
c
c----   Read data and compute average value.
c
        n = 1
 10     continue
          read(10,*,end=20) iv(n,1),iv(n,2), iv(n,3)
```

```
          n = n+1
          goto 10
 20     continue
        nmax = n - 1
        close(10)
c
c----   Find periods.
c
        isum2 = 0
        isum3 = 0
        freq_sum = 0.0
        iavg2sum = 0
        iavg3sum = 0
c
        npoints = 0
        nsteps = 0
        ncycles = 0
        initialized = .false.
        do n = 1, nmax-1
c
c----     Sum up channels 2 and 3, and count points.
c
          npoints = npoints + 1
          isum2 = isum2 + iv(n,2)
          isum3 = isum3 + iv(n,3)
c
c----     Look for the rising step.
c
          if (( iv(n,1) .lt. iv_l ) .and. ( iv(n+1,1) .ge. iv_l )) then
c
c----       If we're already initialised, count the number of steps.
c
            if ( initialized ) then
              nsteps = nsteps + 1
c
c----         If we've counted 128 steps, we have a cycle. compute data.
c
              if ( nsteps .eq. 128 ) then
                ncycles = ncycles + 1
                period = float(npoints) / cal_rate
                freq(ncycles) = 1.0 / period
                iavg2(ncycles) = isum2 / npoints
                iavg3(ncycles) = isum3 / npoints
c
                freq_sum = freq_sum + freq(ncycles)
                iavg2sum = iavg2sum + iavg2(ncycles)
                iavg3sum = iavg3sum + iavg3(ncycles)
c               write(*,*) "-> ",ncycles,npoints,period,freq(ncycles)
c
                nsteps = 0
                npoints = 0
                isum2 = 0
                isum3 = 0
              endif
            else
              initialized = .true.
              npoints = 0
```

```
            isum2 = 0
            isum3 = 0
          endif
        endif
      enddo
c
c----  Convert to units.
c
      freq_avg = freq_sum / ncycles
      drag_fore = iavg2sum / ncycles * force_slope_fore
      drag_aft  = iavg3sum / ncycles * force_slope_aft
c
c----  Compute deviations.
c
      freq_dev = 0.0
      dragf_dev = 0.0
      draga_dev = 0.0
      do n = 1, ncycles
        freq_dev = freq_dev + (freq(n) - freq_avg)**2
        dragf = iavg2(n)*force_slope_fore
        draga = iavg3(n)*force_slope_aft
        dragf_dev = dragf_dev + ( drag_fore - dragf)**2
        draga_dev = draga_dev + ( drag_aft  - draga)**2
      enddo
      freq_dev = sqrt(freq_dev / (ncycles-1))
      dragf_dev = sqrt(dragf_dev / (ncycles-1))
      draga_dev = sqrt(draga_dev / (ncycles-1))
c
      if ( verbose ) then
        write(*,900)
        write(*,901)
        write(*,1000) drag_file, ncycles
        write(*,1001) freq_avg, freq_dev
        write(*,1002) drag_fore, dragf_dev
        write(*,1003) drag_aft, draga_dev
        write(*,*)
      endif
c
  900  format('_____________________________________________________')
  901  format('Zero-offset measurements:')
 1000  format('   File ',a10,': ',i2,' cycles processed')
 1001  format('      Frequency = ',f6.4,' +/- ',f6.4,' Hz')
 1002  format('      Drag (F)  = ',f6.4,' +/- ',f6.4,' N')
 1003  format('      Drag (R)  = ',f6.4,' +/- ',f6.4,' N')
c
      return
      end
```

COMPUTE_THRUST.F

```
      subroutine compute_thrust()
c
      common /dacpar/ cal_rate, thrust_rate, iv_l, iv_h
      common /calibr/ force_slope_fore, force_dev_fore,
     &                force_slope_aft, force_dev_aft
      common /drags / drag_fore, drag_aft, dragf_dev, draga_dev
```

```
       common /thrust/ thrust_fore, thrust_aft, thrustf_dev,
thrusta_dev
       common /freqs / freq_avg, freq_dev
       common /params/ verbose
       common /filenm/ drag_file, thrust_file, vel_file
       character*30 drag_file, thrust_file, vel_file
c
       real freq(100)
       integer iv(500000,3)
       real avg2(100), avg3(100)
       character*10 filename
       logical its_there, initialized, verbose
c
       open(10,file=thrust_file,form='formatted',status='old')
c
c----  Read data and compute average value.
c
       n = 1
 10    continue
         read(10,*,end=20) iv(n,1),iv(n,2), iv(n,3)
         n = n+1
         goto 10
 20    continue
       nmax = n - 1
       close(10)
       do n = 1, nmax, 2
         write(21,*) iv(n,1),iv(n,2), iv(n,3)
       enddo
c
c----  Find periods.
c
       isum2 = 0
       isum3 = 0
       freq_sum = 0.0
       avg2sum = 0
       avg3sum = 0
c
       npoints = 0
       nsteps = 0
       ncycles = 0
       initialized = .false.
       do n = 1, nmax-1
c
c----    Sum up channels 2 and 3, and count points.
c
         npoints = npoints + 1
         isum2 = isum2 + iv(n,2)
         isum3 = isum3 + iv(n,3)
c
c----    Look for the rising step.
c
         if (( iv(n,1) .lt. iv_l ) .and. ( iv(n+1,1) .ge. iv_l )) then
c
c----      If we're already initialised, count the number of steps.
c
           if ( initialized ) then
             nsteps = nsteps + 1
```

```
c
c----         If we've counted 128 steps, we have a cycle. compute data.
c
              if ( nsteps .eq. 128 ) then
                ncycles = ncycles + 1
                period = float(npoints) / thrust_rate
                freq(ncycles) = 1.0 / period
                avg2(ncycles) = float(isum2) / npoints
                avg3(ncycles) = float(isum3) / npoints
c
                freq_sum = freq_sum + freq(ncycles)
                avg2sum = avg2sum + avg2(ncycles)
                avg3sum = avg3sum + avg3(ncycles)
                write(*,999) ncycles,n,npoints,period,freq(ncycles)
 999            format('->',i3,i8,i6,2f12.6)
c
                nsteps = 0
                npoints = 0
                isum2 = 0
                isum3 = 0
              endif
            else
              write(*,*) 'Init:',n
              initialized = .true.
              npoints = 0
              isum2 = 0
              isum3 = 0
            endif
          endif
        enddo
c
c----  Convert to units.
c
        freq_avg = freq_sum / ncycles
        thrust_fore = avg2sum / ncycles * force_slope_fore - drag_fore
        thrust_aft  = avg3sum / ncycles * force_slope_aft - drag_aft
c
c----  Compute deviations.
c
        freq_dev = 0.0
        thrustf_dev = 0.0
        thrusta_dev = 0.0
        do n = 1, ncycles
          freq_dev = freq_dev + (freq(n) - freq_avg)**2
          thrustf = avg2(n)*force_slope_fore - drag_fore
          thrusta = avg3(n)*force_slope_aft  - drag_aft
          thrustf_dev = thrustf_dev + ( thrust_fore - thrustf)**2
          thrusta_dev = thrusta_dev + ( thrust_aft  - thrusta)**2
        enddo
        freq_dev = sqrt(freq_dev / (ncycles-1))
        thrustf_dev = sqrt(thrustf_dev / (ncycles-1))
        thrusta_dev = sqrt(thrusta_dev / (ncycles-1))
c
        if ( verbose ) then
          write(*,900)
          write(*,901)
          write(*,1000) thrust_file, ncycles
```

```
          write(*,1001) freq_avg, freq_dev
          write(*,1002) thrust_fore, thrustf_dev
          write(*,1003) thrust_aft, thrusta_dev
          write(*,*)
        endif
c
  900   format('_____________________________________________________')
  901   format('Thrust measurements:')
 1000   format('   File ',a10,': ',i2,' cycles processed')
 1001   format('      Frequency  = ',f6.4,' +/- ',f6.4,' Hz')
 1002   format('      Thrust (F) = ',f6.4,' +/- ',f6.4,' N')
 1003   format('      Thrust (R) = ',f6.4,' +/- ',f6.4,' N')
c
        return
        end
```

STRIP_COMMENT.F

```
        subroutine strip_comment(iunit)
c
c----   Subroutine strip_comment is used to remove comment lines
c----   from input data. A comment line defined as a line beginning
c----   with '#'.
c
        character*1 one
c
  10    read(iunit,1000) one
        if ( ichar(one) .eq. 35 ) then
          goto 10
        else
          backspace iunit
        endif
c
 1000   format(a1)
c
        return
        end
```

USAGE.F

```
c
c----   Usage stuff.
c
        subroutine usage()
c
        write(*,*)
     & 'Usage: thrust cal_rate data_rate vel_file drag_file
thrust_file'
        write(*,*)
     & '   cal_rate:    samples/second for the zero-offset file'
        write(*,*)
     & '   data_rate:    samples/second for the data file'
        write(*,*)
     & '   vel_file:    file with velocity entries'
        write(*,*)
     & '   drag_file:   file with drag data'
        write(*,*)
```

```
     & '   thrust_file: file with thrust data'
c
       stop
       end
```

LIST OF REFERENCES

1. Fish, F. E., and Rohr, J. J., "Review of Dolphin Hydrodynamics and Swimming Performance," Technical Report 1801, SSC San Diego, August 1999.

2. Triantafyllou, M. S. & Triantafyllou, G. S., 1995 "An Efficient Swimming Machine," *Scientific American*, 272, 64-70.

3. Knoller, R., "Die Gesetze des Luftwiderstandes," Flugund Motortechnik (Wien), Vol. 3, No. 21, 1909, pp. 1-7.

4. Betz, A., "Ein Beitrag zur Erklarung des Segelfluges," Zeitschrift fur Flugtechnik und Motorluftschiffahrt, Vol. 3, January 1912, pp. 269-272.

5. Katzmayr, R., "Effect of Periodic Changes of Angle of Attack on Behavior of Airfoils," NACA Report No. 147, Oct. 1922 (translated from Zeitschrift fur Flugtechnik und Motorluftschiffart, March 31, 1922, pp.80-82, and April 13, 1922, pp. 95-101).

6. Birnbaum, W., "Das ebene Problem des schlagenden Fluels," Zeitschrift fur Angewandte Mathematik und Mechanik, Vol. 4, No. 4, August 1924, pp 277-292.

7. Birnbaum, W., "Der Schlagflugelpropeller und die kleinen Schwingungen elastich befestigter Tragflugel," Zeitschrift fur Flugtechnik und Motorluftschiffahrt, Vol. 15, 1924, pp. 128-134.

8. Jones, K.D. and Platzer, M.F., "On the Design of Efficient Micro Air Vehicles," Design and Nature - Comparing Design in Nature with Science and Engineering Eds. Brebbia, C.A., Sucharov, L.J. and Pascolo, P., WIT Press, Southampton, UK, 2002, pp. 67-76.

9. Von Karman, T. and Burgers, J.M., "General Aerodynamic Theory – Perfect Fluids," Division E, Vol. II, Aerodynamic Theory, Ed. Durand, W.F., 1943, p. 308.

10. Jones, K.D. and Platzer, M.F., "Numerical Computation of Flapping-Wing Propulsion and Power Extraction," AIAA Paper No. 97-0826, *35th AIAA Aerospace Sciences Meeting*, Reno, Nevada, Jan. 1997.

11. Garrick, I.E., "Propulsion of a Flapping and Oscillating Airfoil," NACA Report 567, 1936.

12. Schmidt, W., "Der Wellpropeller, ein neuer Antrieb fuer Wasser-, Land-, und Luftfahrzeuge," Z. Flugwiss. Vol.13, 1965, pp. 472-479.

13. Teng, N., "The Development of a Computer Code (U2DIIF) for the Numerical Solution of Unsteady, Inviscid and Incompressible Flow Over an Airfoil", Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1987.

14. Dohring, Claus M., "Experimental Analysis of the wake of an oscillating airfoil," Master's Thesis, Naval Postgraduate School, Monterey, CA, 1996, p120.

15. Prouty, Raymond W. *Helicopter Performance, Stability, and Control*, PWS Publishers, Boston, 1986, p4.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Max Platzer
   Naval Postgraduate School
   Monterey, California

4. Dr. Kevin Jones
   Naval Postgraduate School
   Monterey, California

5. Kevin Ferguson
   NAS Pensacola
   Pensacola, Florida

6. Chris Bradshaw
   NAS Pensacola
   Pensacola, Florida

7. Richard Paganucci
   Doylestown, Pennsylvania

8. Doug Yuengling
   New Bern, North Carolina